

**OPTIMASI SUSUNAN BAHAN MAKANAN BAGI ANAK
PENDERITA *ATTENTION DEFICIT HYPERACTIVITY DISORDER*
(ADHD) MENGGUNAKAN ALGORITME GENETIKA**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Muhammad Taufan
NIM: 125150200111022



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

OPTIMASI SUSUNAN BAHAN MAKANAN BAGI ANAK PENDERITA *ATTENTION DEFICIT HYPERACTIVITY DISORDER* (ADHD) MENGGUNAKAN ALGORITME GENETIKA

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Muhammad Taufan
NIM: 125150200111022

Skripsi ini telah diuji dan dinyatakan lulus pada
03 Mei 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II


Imam Cholissodin, S.Si, M.Kom
NIK: 201201 850719 1 001


Putra Pandu Adikara, S.Kom, M.Kom
NIP: 19850725 200812 1 002

Mengetahui
Ketua Jurusan Teknik Informatika

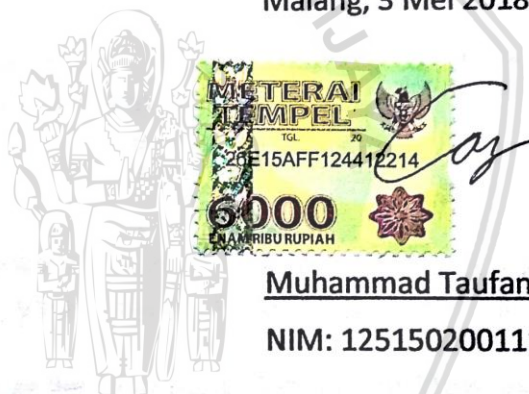

Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 3 Mei 2018



Muhammad Taufan

NIM: 125150200111022

KATA PENGANTAR

Dalam pengerjaan dan penyelesaian skripsi yang berjudul “**OPTIMASI SUSUNAN BAHAN MAKANAN BAGI ANAK PENDERITA ATTENTION DEFICIT HYPERACTIVITY DISORDER (ADHD) MENGGUNAKAN ALGORITME GENETIKA**” sebagai salah satu syarat dalam menyelesaikan Program Sarjana di Fakultas Ilmu Komputer Universitas Brawijaya Malang, penulis banyak mendapat bantuan dan bimbingan dari berbagai pihak, oleh karena itu dalam kesempatan yang baik ini perkenankan penulis untuk mengucapkan rasa terima kasih yang sebesar-besarnya kepada yang terhormat:

1. Bapak **Imam Cholissodin, S.Si, M.Kom.** selaku dosen penasihat akademik sekaligus dosen Pembimbing I yang telah mendorong, membimbing, dan memberikan arahan dari awal perkuliahan, pengerjaan skripsi, hingga terselesaikannya skripsi sebagai akhir dari studi penulis.
2. Bapak **Agus Wahyu Widodo, S.T, M.Cs.** selaku Ketua Program Studi Informatika sekaligus dosen Pembimbing II yang telah mendorong, membimbing, dan memberikan arahan kepada penulis dalam penyelesaian skripsi ini.
3. Bapak **Putra Pandu Adikara, S.Kom, M.Kom.** selaku Pembimbing II menggantikan Pak Wiwid yang sedang cuti dalam tahap pemulihan.
4. Kepada Staf Dosen Fakultas Ilmu Komputer Universitas Brawijaya.
5. Staf karyawan dilingkungan Universitas Brawijaya
6. Kepada Bapak **Mulyadi** dan Mama **Dina Afriyanti** serta seluruh keluarga yang telah memberikan dorongan berupa moril dan materil selama penulis menimba ilmu di FILKOM UB, adik **Sabrina, Xena**, dan **Dhana** yang menjadi semangat penulis untuk menyelesaikan skripsi ini.
7. Sahabat-sahabatku **Alpha 2012** yang menjadi teman seperjuangan.
8. Sahabat-sahabatku **Kontrakan C308** yang menjadi pendorong dalam penyelesaian skripsi ini.
9. Sahabat-sahabatku **BEM Bersatu II** dan **BEM Bersatu III** yang memberikan banyak pengalaman dalam berorganisasi selama masa perkuliahan penulis.

10. Untuk seseorang yang menemani penulis dalam penyelesaian skripsi ini.

11. Kepada keluarga Bapak Mulyadi Banuwijoyo yang memberikan tempat tinggal sementara yang menjadi tempat pengerjaan skripsi ini.

Penulis panjatkan doa semoga Allah SWT memberikan imbalan yang setimpal dan berlipat ganda atas segala bantuan dari semua pihak yang telah membantu penulis dalam mengerjakan dan menyelesaikan skripsi ini. Amin.

Malang, 03 Mei 2018

Penulis

mtaufan1994@gmail.com

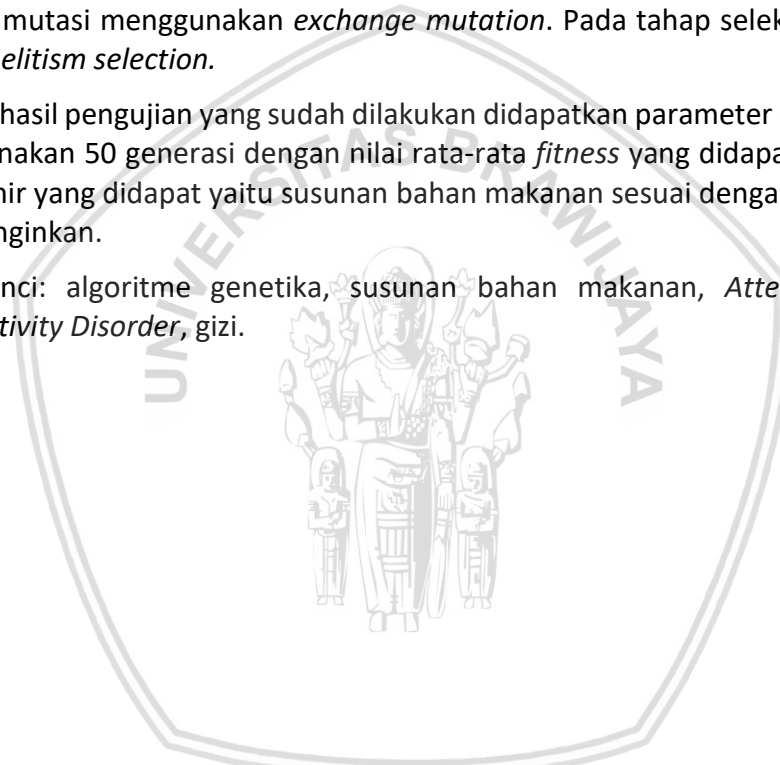


ABSTRAK

Pemberian gizi yang tepat dapat mengurangi gejala *Attention Deficit Hyperactivity Disorder* (ADHD). Akan tetapi anak-anak mudah bosan bila diberikan makanan yang sama secara terus-menerus. Selain itu, harga makanan juga diperhatikan oleh orang tua. Dalam penelitian ini dicari solusi optimal dari permasalahan penyusunan bahan makanan menggunakan algoritme genetika. Dalam penyelesaian masalah ini digunakan representasi integer permutasi dengan panjang kromosom 12 gen pada tiap harinya. Gen-gen tersebut merupakan representasi dari bahan makanan yang terdiri dari karbohidrat, protein, dan lemak. Metode *crossover* yang digunakan adalah *one-cut point*, sedangkan untuk metode mutasi menggunakan *exchange mutation*. Pada tahap seleksi digunakan metode *elitism selection*.

Dari hasil pengujian yang sudah dilakukan didapatkan parameter optimal yaitu menggunakan 50 generasi dengan nilai rata-rata *fitness* yang didapatkan 13.928. Hasil akhir yang didapat yaitu susunan bahan makanan sesuai dengan jumlah hari yang diinginkan.

Kata kunci: algoritme genetika, susunan bahan makanan, *Attention Deficit Hyperactivity Disorder*, gizi.

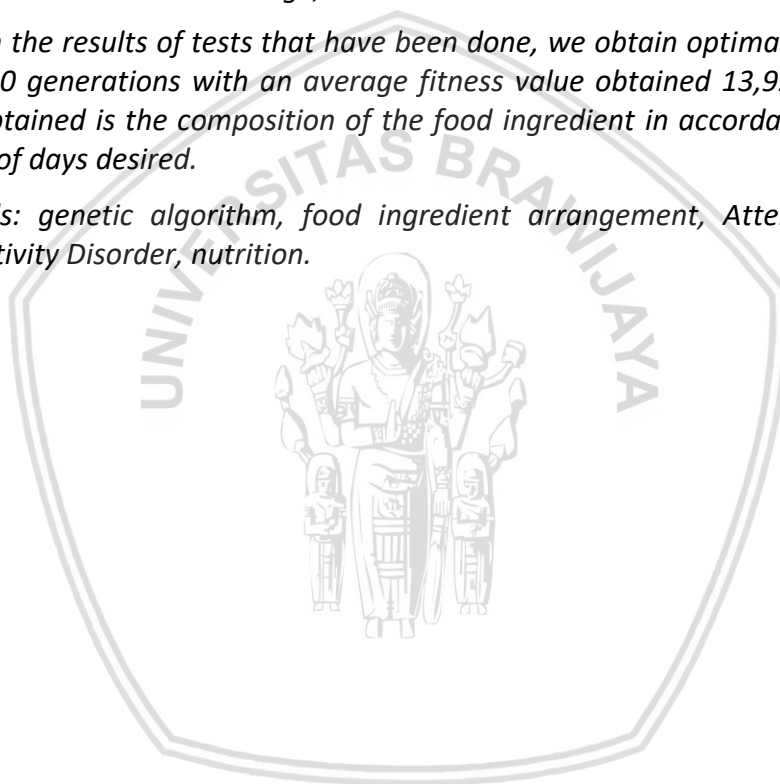


ABSTRACT

Proper nutrition can reduce symptoms of Attention Deficit Hyperactivity Disorder (ADHD). However, children are easily bored when given the same food constantly. In addition, food prices are also considered by parents. In this study sought the optimal solution of the problem of food ingredient preparation using genetic algorithm. To solve this problem, we use an integer permutation representation with chromosome length of 12 genes per day. These genes are a representation of a diet consisting of carbohydrates, proteins, and fats. The crossover method used is one-cut point, while for mutation method using exchange mutation. For the selection stage, elitism selection method is used.

From the results of tests that have been done, we obtain optimal parameters that is 50 generations with an average fitness value obtained 13,928. The final result obtained is the composition of the food ingredient in accordance with the number of days desired.

Keywords: *genetic algorithm, food ingredient arrangement, Attention Deficit Hyperactivity Disorder, nutrition.*



DAFTAR ISI

PERSETUJUAN	Error! Bookmark not defined.
PERNYATAAN ORISINALITAS	Error! Bookmark not defined.
KATA PENGANTAR.....	iv
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
DAFTAR LAMPIRAN	xiv
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan masalah	2
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	4
2.1 Kajian Pustaka	4
2.2 <i>Attention Deficit Hyperactivity Disorder</i> (ADHD).....	6
2.2.1 Pengertian <i>Attention Deficit Hyperactivity Disorder</i> (ADHD)	6
2.2.2 Pembagian Jenis <i>Attention Deficit Hyperactivity Disorder</i> (ADHD)	6
2.2.3 <i>Attention Deficit Hyperactivity Disorder</i> (ADHD) Diet.....	6
2.3 Susunan Bahan Makanan	6
2.3.1 Kalori	6
2.3.2 Karbohidrat	7
2.3.3 Protein.....	7
2.3.4 Lemak	8
2.4 Algoritme Genetika	8
2.4.1 Parameter Algoritme Genetika	8
2.4.2 Siklus Algoritme Genetika	9
BAB 3 METODOLOGI	13

3.1 Studi Pustaka.....	13
3.2 Analisis Kebutuhan	14
3.2.1 Kebutuhan Fungsional.....	14
3.2.2 Kebutuhan Lingkungan Implementasi	14
3.3 Pengumpulan Data	14
3.4 Perancangan Sistem.....	14
3.5 Implementasi Sistem	15
3.6 Pengujian dan Analisis Sistem	15
3.7 Pengambilan Kesimpulan dan Saran	15
BAB 4 PERANCANGAN.....	16
4.1 Formulasi Permasalahan.....	16
4.1.1 Deskripsi Masalah	16
4.1.2 Daftar Bahan Makanan	16
4.1.3 Perhitungan Kebutuhan Gizi	16
4.2 Tahap-tahap Penyelesaian Masalah menggunakan Algoritme Genetika	18
4.2.1 Representasi Kromosom	19
4.2.2 Inisialisasi Populasi Awal	20
4.2.3 Reproduksi	21
4.2.4 Evaluasi.....	27
4.3 Perancangan Antarmuka	40
4.3.1 <i>Input</i> Data Anak.....	40
4.3.2 Menu Perhitungan Algoritme Genetika.....	41
4.3.3 Menu Hasil Optimasi	42
BAB 5 IMPLEMENTASI	43
5.1 Implementasi Algoritme Genetika.....	43
5.1.1 Proses <i>Input</i>	43
5.1.2 Proses Representasi Kromosom	44
5.1.3 <i>Crossover</i>	45
5.1.4 Mutasi	46
5.2 Implementasi Antarmuka	48
BAB 6 PENGUJIAN DAN PEMBAHASAN.....	51

6.1 Pengujian Jumlah Populasi	51
6.2 Pengujian Kombinasi <i>Cr</i> dan <i>Mr</i>	52
6.3 Pengujian Jumlah Generasi.....	53
6.4 Analisis Global.....	55
BAB 7 PENUTUP	56
7.1 Kesimpulan.....	56
7.2 Saran	57
DAFTAR PUSTAKA.....	58
LAMPIRAN A DAFTAR BAHAN MAKANAN.....	59
LAMPIRAN B PENGUJIAN	62



DAFTAR TABEL

Tabel 2.1 Kajian Pustaka	4
Tabel 4.1 Data Anak	17
Tabel 4.2 Kebutuhan Zat Gizi	18
Tabel 4.3 Anjuran Porsi Sehari	18
Tabel 4.4 Porsi Kebutuhan Gizi	18
Tabel 4.5 Penjelasan Kromosom	19
Tabel 4.6 Populasi Awal	20
Tabel 4.7 Proses <i>Crossover</i>	22
Tabel 4.8 <i>Exchange Mutation</i>	24
Tabel 4.9 Hasil Konversi Gen	30
Tabel 4.10 Berat dan Nilai Gizi	33
Tabel 4.11 Penalti Gizi	35
Tabel 4.12 Harga Bahan Makanan	36
Tabel 4.13 Variasi Bahan Makanan	37
Tabel 4.14 Populasi <i>Parent</i> dan <i>Offspring</i>	39
Tabel 4.15 Urutan Populasi Berdasarkan <i>Fitness</i>	39
Tabel 4.16 Individu Generasi Baru	40
Tabel 6.1 Hasil Pengujian Jumlah Populasi	51
Tabel 6.2 Hasil Pengujian Kombinasi <i>Cr</i> dan <i>Mr</i>	52
Tabel 6.3 Hasil Pengujian Jumlah Generasi	54

DAFTAR GAMBAR

Gambar 2.1 Contoh Proses <i>Crossover</i>	10
Gambar 2.2 Contoh Proses <i>Mutation</i>	10
Gambar 3.1 Blok Diagram Metodologi Penelitian	13
Gambar 4.1 Inisialisasi Populasi Awal	20
Gambar 4.2 <i>Flowchart Crossover</i>	21
Gambar 4.3 Proses Pemilihan <i>Parent</i>	23
Gambar 4.4 <i>Flowchart One-cut Point Crossover</i>	24
Gambar 4.5 <i>Flowchart</i> Penentuan Posisi Gen Mutasi	25
Gambar 4.6 <i>Flowchart</i> Pemilihan <i>Parent</i>	26
Gambar 4.7 <i>Flowchart</i> Pertukaran Gen	27
Gambar 4.8 <i>Flowchart</i> Evaluasi	28
Gambar 4.9 <i>Flowchart</i> Konversi Gen	29
Gambar 4.10 <i>Flowchart</i> Hitung Berat Bahan Makanan	31
Gambar 4.11 <i>Flowchart</i> Perhitungan Nilai Gizi	32
Gambar 4.12 <i>Flowchart</i> Perhitungan Penalti Gizi	34
Gambar 4.13 <i>Flowchart</i> Menghitung Harga Bahan Makanan	35
Gambar 4.14 <i>Flowchart</i> Menghitung Variasi Bahan Makanan	37
Gambar 4.15 <i>Flowchart</i> Proses Seleksi	38
Gambar 4.16 <i>Input</i> Data Anak	41
Gambar 4.17 <i>Input</i> Parameter Algoritme Genetika	41
Gambar 4.18 Perhitungan Algoritme Genetika	42
Gambar 4.19 Menu Hasil Optimasi	42
Gambar 5.1 <i>Source Code</i> Proses <i>Input</i>	43
Gambar 5.2 <i>Source Code</i> Representasi Kromosom	44
Gambar 5.3 <i>Source Code Crossover</i>	45
Gambar 5.4 <i>Source Code</i> Mutasi	47
Gambar 5.5 Tampilan Bagian Pertama <i>Input</i> Data Anak	48
Gambar 5.6 Tampilan Bagian Kedua <i>Input</i> Parameter	49
Gambar 5.7 Tampilan Bagian Kedua Proses Optimasi	49
Gambar 5.8 Tampilan Bagian Ketiga Hasil Optimasi	50

Gambar 6.1 Grafik Hasil Pengujian Jumlah Populasi	52
Gambar 6.2 Grafik Pengujian Kombinasi Cr dan Mr	53
Gambar 6.3 Grafik Pengujian Jumlah Generasi	54



DAFTAR LAMPIRAN

LAMPIRAN A DAFTAR BAHAN MAKANAN.....	59
A.1 Bahan Makanan Sumber Karbohidrat	59
A.2 Bahan Makanan Sumber Protein Hewani	60
A.3 Bahan Makanan Sumber Protein Nabati.....	61
A.4 Bahan Makanan Sumber Lemak.....	61
LAMPIRAN B PENGUJIAN	62
B.1 Tabel Pengujian Jumlah Populasi	62
B.2 Tabel Pengujian Kombinasi <i>Cr</i> dan <i>Mr</i>	63
B.3 Tabel Pengujian Jumlah Generasi.....	64



BAB 1 PENDAHULUAN

1.1 Latar belakang

Semua makhluk hidup membutuhkan makanan, begitu pula manusia. Banyak sekali jenis makanan yang biasa dikonsumsi oleh manusia sehari-hari. Jenis-jenis makanan tersebut memiliki kandungan zat gizi yang berbeda-beda. Tubuh manusia memerlukan zat-zat gizi seperti karbohidrat, lemak, protein, mineral, dan vitamin agar mampu berfungsi normal (Almatsier, 2009). Makanan yang sehat dan bergizi mampu memberikan zat-zat gizi yang dibutuhkan tersebut. Akan tetapi, besaran zat-zat gizi yang dibutuhkan tiap manusia tidak selalu sama satu dengan lainnya. Terdapat beberapa perbedaan kebutuhan besaran zat-zat gizi yang dibutuhkan manusia sesuai dengan keadaannya, misalnya penyakit yang diderita.

Banyak sekali jenis penyakit yang mungkin diderita oleh manusia. Salah satunya adalah *Attention Deficit Hyperactivity Disorder* (ADHD). Pada DSM-V dijelaskan bahwa ADHD adalah kelainan pada pertumbuhan saraf yang dibagi menjadi beberapa bagian yaitu kesulitan dalam pemusatan perhatian, semauanya sendiri, dan tidak bisa diam serta impulsif. Penyakit ini bisa dideteksi sejak dini yaitu dimasa anak-anak. Hingga saat ini ADHD belum bisa disembuhkan, akan tetapi dapat diminimalkan dengan pemberian terapi. Salah satu terapi ADHD adalah melalui pemberian asupan makanan. Anak-anak mudah bosan bila diberikan makanan yang sama secara terus-menerus. Ini menimbulkan masalah baru yaitu bagaimana membuat susunan bahan makanan yang sesuai untuk anak penderita ADHD yang bervariasi. Selain itu, harga makanan juga menjadi pertimbangan bagi orang tua dalam penyediaan makanan.

Metode *heuristic* yang didasari mekanisme evolusi biologis mampu menyelesaikan permasalahan penyusunan bahan makanan secara optimal untuk anak penderita ADHD tersebut. Salah satu metode yang bisa digunakan adalah algoritme genetika. Algoritme genetika adalah algoritme optimasi yang terinspirasi dari proses seleksi alam pada makhluk hidup, dimana individu dengan gen terbaiklah yang akan bertahan sampai akhir dan dapat meneruskan keturunannya. Pada penelitian sebelumnya algoritme genetika digunakan untuk menyelesaikan masalah penentuan kelembaban tanah optimum untuk budidaya padi sawah SRI (*System of Rice Intensification*) (Arif et al., 2014). Penerapan algoritme genetika juga ditemukan dalam optimasi distribusi barang (Panhares & Mahmudy, 2015). Ada pula penerapan algoritme genetika yang berhubungan dengan permasalahan komposisi makanan dalam optimasi biaya pemenuhan gizi dan nutrisi pada manusia lanjut usia (Suci, Mahmudy, & Putri, 2015).

Dilihat dari penelitian-penelitian sebelumnya, bisa ditarik kesimpulan bahwa algoritme genetika mampu memberikan solusi untuk permasalahan-permasalahan optimasi yang kompleks. Oleh sebab itu, penulis memiliki gagasan untuk melakukan penelitian tentang optimasi susunan bahan makanan bagi anak penderita *Attention Deficit Hyperactivity Disorder* (ADHD) menggunakan algoritme genetika.

1.2 Rumusan masalah

Berdasarkan uraian latar belakang di atas, maka dapat dirumuskan permasalahan pada tugas akhir ini yaitu sebagai berikut:

1. Bagaimana menerapkan algoritme genetika pada susunan bahan makanan bagi anak penderita *Attention Deficit Hyperactivity Disorder* (ADHD)?
2. Bagaimana kualitas solusi algoritme genetika terhadap penentuan susunan bahan makanan bagi anak penderita *Attention Deficit Hyperactivity Disorder* (ADHD)?

1.3 Tujuan

Tujuan utama dari penelitian ini yaitu sebagai berikut:

1. Menerapkan algoritme genetika pada susunan bahan makanan bagi anak penderita *Attention Deficit Hyperactivity Disorder* (ADHD).
2. Mengetahui kualitas solusi algoritme genetika terhadap penentuan susunan bahan makanan bagi anak penderita *Attention Deficit Hyperactivity Disorder* (ADHD).

1.4 Manfaat

Penelitian ini diharapkan dapat bermanfaat untuk berbagai pihak. Manfaat dari penelitian ini adalah sebagai berikut:

Manfaat bagi peneliti:

1. Dapat menggali pengetahuan dalam pengaplikasian algoritme genetika.
2. Mendapatkan kesempatan untuk menerapkan penelitian di kehidupan nyata.

Manfaat Bagi Masyarakat:

1. Sebagai rekomendasi bagi orang tua dalam menentukan bahan makanan yang bervariasi dan memenuhi kebutuhan gizi bagi anak penderita *Attention Deficit Hyperactivity Disorder* (ADHD).
2. Mendapatkan pengetahuan dari membaca penelitian yang dibuat.

1.5 Batasan masalah

Agar permasalahan yang dirumuskan dapat lebih terfokus, maka penelitian tugas akhir ini dibatasi dalam hal:

1. *Attention Deficit Hyperactivity Disorder* (ADHD) adalah berbagai gejala perilaku yang dapat meliputi hiperaktif, kurangnya perhatian atau konsentrasi dan kecenderungan perilaku impulsif. Perilaku impulsif pada anak meliputi sulit mengontrol reaksi dan seringkali bertindak tanpa memikirkan akibatnya.

2. Penderita *Attention Deficit Hyperactivity Disorder* (ADHD) adalah anak-anak dengan rentang usia 2 sampai 6 tahun.

1.6 Sistematika pembahasan

Sistematika penulisan laporan tugas akhir ini adalah sebagai berikut:

BAB I Pendahuluan

Memuat latar belakang permasalahan, identifikasi dan pembatasan masalah, rumusan masalah, tujuan dan manfaat, dan sistematika penulisan.

BAB II Landasan Kepustakaan

Menjelaskan komponen-komponen utama yang dipakai pada optimasi susunan bahan makanan bagi anak penderita *Attention Deficit Hyperactivity Disorder* (ADHD) menggunakan algoritme genetika.

BAB III Metodologi

Membahas metode yang digunakan dalam penelitian yang terdiri dari studi pustaka, analisis kebutuhan, pengumpulan data, perancangan sistem, implementasi sistem, pengujian dan analisis sistem, serta pengambilan kesimpulan dan saran.

BAB IV Perancangan

Menguraikan perancangan sistem optimasi susunan bahan makanan bagi anak penderita *Attention Deficit Hyperactivity Disorder* (ADHD) dengan menggunakan algoritme genetika.

BAB V Implementasi

Membahas mengenai implementasi dari *software* sesuai dengan yang sudah dirancang sebelumnya pada bab perancangan.

BAB VI Pengujian dan Analisis

Membahas mengenai proses dan hasil dari pengujian sistem yang telah dibuat dan menganalisis apakah sistem sudah berjalan sesuai dengan yang sudah dirancang sebelumnya.

BAB VII Penutup

Berisi kesimpulan yang didapatkan setelah melalui rangkaian proses pada bab-bab sebelumnya, dan juga saran-saran untuk pengembangan selanjutnya dari optimasi susunan bahan makanan bagi anak penderita *Attention Deficit Hyperactivity Disorder* (ADHD) dengan menggunakan algoritme genetika.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Kajian pustaka membahas tentang perbandingan antara penelitian ini dengan tiga penelitian terdahulu. Penelitian pertama yang dijadikan acuan adalah penelitian yang dilakukan oleh Panhares & Mahmudy (2015) yang membahas mengenai optimasi distribusi barang menggunakan algoritme genetika. Penelitian kedua adalah penelitian yang dilakukan oleh Arif, Setiawan, & Mizoguchi (2014) yang membahas tentang penentuan kelembaban tanah optimum untuk budidaya padi sawah SRI (*System of Rice Intensification*) menggunakan algoritme genetika. Penelitian ketiga adalah penelitian yang dilakukan oleh Suci, Mahmudy, & Putri (2015) yang membahas tentang optimasi biaya pemenuhan gizi dan nutrisi pada manusia lanjut usia menggunakan algoritme genetika.

Pada penelitian pertama, metode *crossover* yang digunakan adalah *one cut-point crossover* dan metode *mutation* yang digunakan adalah *reprocal exchange mutation*. Pada proses seleksi digunakan metode *elitism*. Dari penelitian tersebut, dapat disimpulkan bahwa algoritme genetika mampu menyelesaikan permasalahan tersebut.

Pada penelitian kedua, metode *crossover* dan metode *mutation* tidak dijelaskan secara rinci. Namun, dapat diketahui metode seleksi yang digunakan adalah metode *elitism*. Dari penelitian tersebut, algoritme genetika mampu memberikan solusi yang lebih baik.

Pada penelitian ketiga, metode *crossover* yang digunakan adalah *one-cut point crossover* dan metode *mutation* yang digunakan adalah *reprocal exchange mutation*. Metode seleksi yang digunakan adalah metode *elitism*. Dari penelitian tersebut bisa ditarik kesimpulan bahwa algoritme genetika mampu menyelesaikan permasalahan biaya pemenuhan gizi dan nutrisi dengan memberikan rekomendasi bahan makanan dalam sehari yang memiliki harga minimum.

Tabel 2.1 Kajian Pustaka

No.	Judul Penelitian	Objek	Metode yang digunakan	Hasil dan pengujian
		Input dan Parameter		
1.	Optimasi Distribusi Barang	-Data jarak, biaya angkut truk muatan kosong, biaya angkut truk muatan penuh, muatan yang diangkut truk, muatan maksimal truk	Algoritma Genetika	Hasil berupa kombinasi distribusi barang dengan biaya yang minimum.

		-Parameter algoritme genetika (ukuran populasi, generasi, <i>crossover rate</i> , dan <i>mutation rate</i>).		
2.	Penentuan Kelembaban Tanah Optimum Untuk Budidaya Padi Sawah SRI (<i>System of Rice Intensification</i>) Menggunakan Algoritme Genetika	- Data estimasi produksi, perhitungan produktifitas air -Parameter algoritme genetika (ukuran populasi, generasi, <i>crossover rate</i> , dan <i>mutation rate</i>).	Algoritme Genetika	Hasil berupa kombinasi kelembaban tanah yang optimum.
3.	Optimasi Biaya Pemenuhan Gizi dan Nutrisi pada Manusia Lanjut Usia Menggunakan Algoritme Genetika	-Data berat badan dan jenis kelamin. - Parameter algoritme genetika (ukuran populasi, generasi, <i>crossover rate</i> , dan <i>mutation rate</i>).	Algoritme Genetika	Hasil berupa rekomendasi bahan makanan dalam satu hari untuk manusia lanjut usia dengan biaya minimal dan kandungan gizi yang optimal.
4.	Optimasi Susunan Bahan Makanan bagi Anak Penderita <i>Attention Deficit Hyperactivity Disorder</i> (ADHD) Menggunakan Algoritme Genetika	- Alergi Anak - Parameter algoritme genetika (ukuran populasi, generasi, <i>crossover rate</i> , dan <i>mutation rate</i>).	Algoritme Genetika	Hasil berupa rekomendasi susunan bahan makanan dalam satu hari sesuai dengan alergen dan jenis ADHD yang dimasukkan.

Dari ketiga penelitian diatas, penulis menyimpulkan bahwa algoritme genetika mampu menyelesaikan permasalahan optimasi susunan bahan makanan. Dengan demikian, penulis menggunakan algoritme genetika untuk menyelesaikan permasalahan dalam penelitian skripsi ini.

2.2 Attention Deficit Hyperactivity Disorder (ADHD)

Sub bab ini menjelaskan hal-hal yang berhubungan dengan *Attention Deficit Hyperactivity Disorder* (ADHD), yaitu pengertian, pembagian jenis ADHD, dan *diet*.

2.2.1 Pengertian Attention Deficit Hyperactivity Disorder (ADHD)

DSM-V menjelaskan bahwa *Attention Deficit Hyperactivity Disorder* (ADHD) adalah kelainan pada pertumbuhan saraf yang dibagi menjadi beberapa bagian yaitu kesulitan dalam pemusatan perhatian, semaunya sendiri, dan tidak bisa diam serta impulsif. Anak impulsif memiliki kesulitan dalam mengendalikan reaksinya dan seringkali melakukan sesuatu tanpa memikirkan akibatnya. Hingga saat ini ADHD tidak bisa diobati, akan tetapi dapat dikurangi gejalanya dengan cara terapi.

2.2.2 Pembagian Jenis Attention Deficit Hyperactivity Disorder (ADHD)

Berdasarkan DSM-V, ADHD dibagi menjadi beberapa jenis, yaitu *inattention*, *hyperactivity*, dan *impulsive*. Seorang anak dikatakan menderita ADHD ketika memiliki satu atau lebih dari ketiga jenis yang ada.

2.2.3 Attention Deficit Hyperactivity Disorder (ADHD) Diet

Salah satu *diet* bagi anak penderita ADHD adalah *Feingold diet*. Anak penderita ADHD dianjurkan untuk menjauhi makanan yang mengandung pemanis buatan, perasa sintetis, maupun pengawet makanan. Selain itu juga dianjurkan untuk menjauhi makanan yang mengandung *Salicylate*, contohnya apel, jeruk, anggur, teh, dan tomat (Jacobson & Schardt, 1999).

2.3 Susunan Bahan Makanan

Manusia membutuhkan makanan agar tubuh mampu berfungsi dengan normal. Bahan makanan dapat dikatakan lengkap ketika memenuhi syarat 4 sehat 5 sempurna. Bahan makanan harus mengandung kalori, karbohidrat, protein, dan lemak.

2.3.1 Kalori

Kalori atau sumber energi diperlukan manusia dalam melakukan aktivitas fisik sehari-hari. Kalori dapat ditemukan dalam makanan yang mengandung karbohidrat, protein, dan lemak. Kalori sangat penting bagi tubuh untuk menjalankan proses dalam tubuh (Almatsier, 2009). Hampir seluruh sel dalam tubuh manusia melakukan aktivitas secara konstan.

Kebutuhan kalori pada setiap manusia berbeda-beda. Hal ini dipengaruhi oleh tingkat aktivitas, energi yang dikeluarkan, dan ukuran tubuh dari individu tersebut. Faktor terbesar yang mempengaruhi energi yang dikeluarkan adalah proses metabolisme basal (Almatsier, 2009).

Menurut Harris J (1919), Persamaan untuk menghitung berat badan ideal dan kebutuhan kalori ditunjukkan pada Persamaan 2.1, 2.2, dan 2.3 berikut ini:

$$BBI_{Anak} = (U * 2) + 8 \quad (2.1)$$

$$TEE_{Laki-laki} = (65 + (13,7 * BBI) + (5 * TB) - (6,8 * U)) * FA \quad (2.2)$$

$$TEE_{Perempuan} = (655 + (9,6 * BBI) + (1,8 * TB) - (4,7 * U)) * FA \quad (2.3)$$

Keterangan:

TEE : *Total Energy Expenditure* (Total Kebutuhan Energi Sehari)

BBI : Berat Badan Ideal

TB : Tinggi Badan

U : Usia

FA : Faktor Aktivitas

2.3.2 Karbohidrat

Karbohidrat adalah sumber energi utama yang dibutuhkan manusia yang juga relatif mudah didapatkan (Almatsier, 2009). Karbohidrat hanya bersumber dari tumbuh-tumbuhan saja. Tumbuhan memproduksi karbohidrat melalui proses fotosintesis.

Persamaan untuk menghitung kebutuhan karbohidrat harian menurut Hartono (2006) ditunjukkan pada Persamaan 2.4 berikut ini:

$$KK = \frac{r_i * TEE}{4} \quad (2.4)$$

Keterangan:

KK : Kebutuhan Karbohidrat

TEE : *Total Energy Expenditure* (Total Kebutuhan Energi Sehari)

r : Prosentase Kebutuhan Karbohidrat

i : Status Kesehatan

2.3.3 Protein

Protein tersusun dari asam amino. Protein dapat ditemukan diseluruh makhluk hidup. Sumber protein bagi manusia dibagi menjadi dua, yaitu sumber protein hewani dan sumber protein nabati (Almatsier, 2009). Protein hewani bersumber dari hewan, sedangkan protein nabati bersumber dari tumbuhan.

Persamaan untuk menghitung kebutuhan karbohidrat harian menurut Hartono (2006) ditunjukkan pada Persamaan 2.5 berikut ini:

$$KP = \frac{r_i * TEE}{4} \quad (2.5)$$

Keterangan:

- KP : Kebutuhan Protein
 TEE : *Total Energy Expenditure* (Total Kebutuhan Energi Sehari)
 r : Prosentase Kebutuhan Protein
 i : Status Kesehatan

2.3.4 Lemak

Lemak pada manusia biasa dikategorikan sebagai cadangan enegi. Kebutuhan lemak manusia hanyalah berkisar 20-30% dari kebutuhan energi total (Almatsier, 2009).

Persamaan untuk menghitung kebutuhan lemak harian menurut Hartono (2006) dtunjukkan pada Persamaan 2.6 berikut ini:

$$KL = \frac{r_i * TEE}{9} \quad (2.6)$$

Keterangan:

- KL : Kebutuhan Lemak
 TEE : *Total Energy Expenditure* (Total Kebutuhan Energi Sehari)
 r : Prosentase Kebutuhan Lemak
 i : Status Kesehatan

2.4 Algoritme Genetika

Salah satu metode yang bisa digunakan dalam menyelesaikan permasalahan optimasi adalah algoritme genetika. Metode ini didasari oleh proses evolusi alami makhluk hidup dimana individu terbaiklah yang akan terus bertahan. Algoritme genetika ditemukan pada tahun 1970 oleh John Holland dari Universitas Michigan. Terdapat beberapa parameter dalam penyelesaian masalah menggunakan algoritme genetika yang akan dijelaskan selanjutnya.

2.4.1 Parameter Algoritme Genetika

Berikut adalah parameter yang digunakan dalam algoritme genetika:

1. Ukuran Populasi (*PopSize*)

Ukuran populasi berpengaruh pada keefektifan proses algoritme genetika dalam penyelesaian masalah. Ukuran populasi yang digunakan sebisa mungkin dibuat tidak terlalu banyak atau pun terlalu sedikit dimana akan mengurangi hasil dari algoritme genetika dalam pencarian solusi terbaik.

2. Crossover Rate (pc)

Crossover rate (pc) adalah sebuah nilai yang menyatakan perbandingan antara jumlah keturunan yang dihasilkan pada proses *crossover* terhadap *PopSize*. Besarnya pc akan mempengaruhi jumlah keturunan dimana jumlahnya ditentukan dengan rumus $pc \times PopSize$. Semakin besar nilai pc , maka keturunan yang dihasilkan akan semakin banyak. Semakin kecil nilai pc , maka keturunan yang dihasilkan akan semakin sedikit.

3. Mutation Rate (pm)

Mutation rate (pm) adalah sebuah nilai yang menyatakan perbandingan antara jumlah keturunan yang dihasilkan pada proses *mutation* terhadap *PopSize*. Besarnya pm sama halnya dengan pc mempengaruhi jumlah keturunan.

4. Jumlah Generasi

Jumlah generasi adalah sebuah nilai yang menyatakan banyaknya perulangan pada proses algoritme genetika. Nilainya harus ditentukan dengan tepat sehingga hasil yang didapatkan bisa optimal.

2.4.2 Siklus Algoritme Genetika

Terdapat tahapan-tahapan yang harus dikerjakan untuk menyelesaikan masalah menggunakan algoritme genetika yang akan dijelaskan dibawah ini.

2.4.2.1 Representasi Kromosom

Representasi kromosom merupakan proses pembangkitan solusi permasalahan nyata ke dalam kromosom. Representasi kromosom memiliki beberapa jenis yang akan disesuaikan dengan permasalahan, karena dengan representasi yang sesuai akan memudahkan proses penyelesaian masalah menggunakan algoritme genetika.

2.4.2.2 Inisialisasi

Proses inisialisasi dilakukan untuk mendefinisikan individu-individu yang akan dimasukkan kedalam populasi secara acak. Jumlah individu yang diambil akan disesuaikan dengan ukuran populasi atau *popSize* (Mahmudy, 2016).

2.4.2.3 Reproduksi

Proses reproduksi dilakukan untuk menghasilkan individu baru. Individu baru tersebut merupakan keturunan dari individu-individu yang ada di populasi (Mahmudy, 2016). Keturunan dihasilkan melalui proses *crossover* dan *mutation*.

Proses *crossover* memerlukan dua individu yang disebut *parent* untuk menghasilkan keturunan. *Parent* diambil secara acak dari populasi. Masing-masing kromosom *parent* dipotong menjadi dua segmen yang selanjutnya akan ditukar antar *parent* sehingga menghasilkan individu baru atau keturunan. Selain itu ada juga *Crossover rate* yang akan menentukan jumlah keturunan yang perlu dihasilkan. Tiap proses *crossover* akan menghasilkan maksimal dua individu baru. Terdapat beberapa metode dalam proses *crossover*, yaitu *one-cut point crossover*,

position-based crossover, *order-based crossover*, *partial mapped crossover (PMX)*, *order crossover (OX)*, *cycle crossover*, dan *heuristic crossover* (Gen & Cheng dalam Mahmudy, 2016). Contoh penggunaan metode *one cut point*:

	Kromosom														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P2	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

Dipilih satu pasang *parent* P1 dan P2 dengan *cutpoint* pada kromosom ke 10, maka *offspring* yang dihasilkan:

	Kromosom														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
C1	1	2	3	4	5	6	7	8	9	10	26	27	28	29	30
C2	16	17	18	19	20	21	22	23	24	25	11	12	13	14	15

Gambar 2.1 Contoh Proses Crossover

Proses *mutation* hanya memerlukan satu individu yang diambil secara acak dari populasi. Terdapat beberapa metode dalam proses *mutation*, antara lain *reciprocal exchange mutation* dan *insertion mutation*. Pada *reciprocal exchange mutation* akan dipilih dua posisi secara acak yang nilainya akan ditukar satu sama lain untuk menghasilkan individu baru. Pada *insertion mutation* akan dipilih sebuah nilai secara acak yang akan disisipkan pada posisi yang dipilih secara acak. Berikut ini contoh proses *mutation*:

P1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
----	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----

Menukar nilai pada kromosom ke 5 dengan kromosom ke 11

C3	1	2	3	4	11	5	6	7	8	9	10	12	13	14	15
----	---	---	---	---	----	---	---	---	---	---	----	----	----	----	----

Gambar 2.2 Contoh Proses Mutation

2.4.2.4 Evaluasi

Proses evaluasi dilakukan untuk menghitung nilai *fitness* tiap individu. Semakin besar nilai *fitness* maka kemungkinan individu tersebut untuk menjadi solusi semakin besar (Mahmudy, 2016). Nilai *fitness* didapatkan dari Persamaan 2.7.

$$Fitness = f(x) \quad (2.7)$$

$$Fitness = C - f(x) \quad (2.8)$$

$$Fitness = \frac{c}{f(x) + \varepsilon} \quad (2.9)$$

Keterangan:

C = bilangan konstanta

ε = bilangan kecil untuk menghindari pembagian dengan nol

Persamaan 2.7, nilai *fitness* dihitung secara langsung ketika permasalahan memiliki solusi yang bernilai maksimum. Sedangkan Persamaan 2.8 dan 2.9 digunakan ketika permasalahan memiliki solusi yang bernilai minimum.

2.4.2.5 Seleksi

Proses seleksi dilakukan untuk menentukan individu yang dipertahankan untuk melakukan proses reproduksi pada generasi berikutnya sebagai *parent*. Pada umumnya individu dengan nilai *fitness* tertinggi lah yang akan terpilih. Adapun beberapa metode seleksi yang mungkin digunakan.

1. Seleksi *tournament*

Metode seleksi ini dilakukan dengan memilih beberapa individu secara acak yang selanjutnya akan dibandingkan nilai *fitness*-nya. Individu dengan nilai *fitness* tertinggi akan dipilih untuk melanjutkan kembali proses reproduksi pada generasi berikutnya.

2. Seleksi *ranking*

Metode seleksi ini dimulai dengan mengurutkan seluruh individu yang ada dalam populasi berdasarkan nilai *fitness*-nya. Selanjutnya individu dengan nilai *fitness* terendah akan diberi nilai *fitness* baru yang bernilai 1, individu dengan nilai *fitness* terendah kedua diberi nilai 2, dan seterusnya sampai dengan individu dengan nilai *fitness* tertinggi.

3. Seleksi *elitism*

Metode seleksi ini memilih individu dengan nilai *fitness* tertinggi pada setiap generasi untuk melanjutkan proses reproduksi pada generasi berikutnya.

4. Seleksi *roulette wheel*

Metode ini merepresentasikan individu-individu pada populasi sebagai bagian dari sebuah roda dimana ukuran tiap bagian ditentukan oleh besarnya probabilitas seleksi dalam Persamaan 2.10.

$$prob_k = \frac{fitness(P_k)}{totalFitness} \quad (2.10)$$

Keterangan: $fitness(P_k)$ = nilai *fitness* individu ke- k

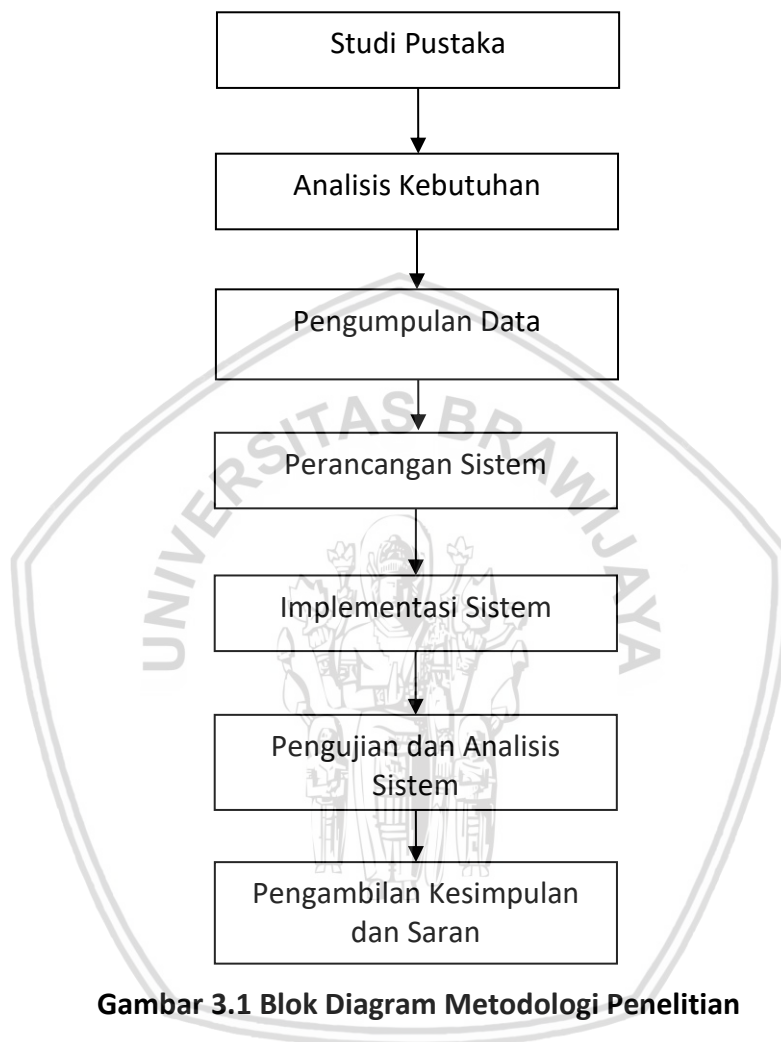
$totalFitness$ = total nilai *fitness* populasi

Semakin tinggi nilai *fitness* sebuah individu, maka semakin besar ukuran bagiannya yang juga meningkatkan kemungkinan individu tersebut untuk terpilih kembali.



BAB 3 METODOLOGI

Metodologi penelitian yang dilakukan dalam penelitian ini melalui beberapa tahapan-tahapan yang dapat diilustrasikan dengan diagram blok metodologi penelitian seperti pada Gambar 3.1:



Gambar 3.1 Blok Diagram Metodologi Penelitian

3.1 Studi Pustaka

Studi pustaka digunakan untuk mendapatkan referensi yang dapat digunakan sebagai dasar teori dari beberapa penelitian yang sudah dilakukan sebelumnya yang membantu dalam penulisan tugas akhir ini. Adapun beberapa teori yang membantu penelitian ini adalah:

1. Makanan sehat dan bergizi
2. Zat gizi makro
3. Gizi bagi penderita *Attention Deficit Hyperactivity Disorder* (ADHD)
4. Metode optimasi
5. Algoritme evolusi

6. Algoritme genetika

3.2 Analisis Kebutuhan

Analisa kebutuhan digunakan untuk menganalisis hal-hal yang dibutuhkan dalam membangun sistem optimasi susunan bahan makanan bagi penderita *Attention Deficit Hyperactivity Disorder* (ADHD) menggunakan algoritme genetika.

3.2.1 Kebutuhan Fungsional

Kebutuhan fungsional sistem meliputi:

1. Sistem mampu melakukan optimasi susunan bahan makanan bagi anak penderita *Attention Deficit Hyperactivity Disorder* (ADHD) menggunakan algoritme genetika.
2. Sistem mampu menghasilkan *output* berupa bahan makanan untuk anak penderita *Attention Deficit Hyperactivity Disorder* (ADHD) beserta takaran masing-masing bahan makanan.

3.2.2 Kebutuhan Lingkungan Implementasi

Spesifikasi lingkungan implementasi dalam pembuatan sistem meliputi:

1. Spesifikasi kebutuhan *software*:
 - Windows 10 64 bit sebagai sistem operasi yang digunakan.
 - Microsoft Word 2016 digunakan untuk dokumentasi penulisan.
 - Microsoft Excel 2016 digunakan untuk dokumentasi manualisasi perhitungan.
 - Javascript sebagai bahasa pemrograman yang digunakan untuk pengembangan sistem.
 - Atom sebagai lingkungan pengembangan sistem.
2. Spesifikasi kebutuhan *hardware*:
 - Prosesor Intel® Core™ i5-6200U 2.40GHz.
 - RAM 4 GB.
 - *Hardisk* dengan kapasitas 500 GB
 - Monitor 14'

3.3 Pengumpulan Data

Pengumpulan data dilakukan dengan cara wawancara atau permintaan data set kepada pakar gizi yang memiliki pengetahuan tentang gizi bagi anak penderita *Attention Deficit Hyperactivity Disorder* (ADHD).

3.4 Perancangan Sistem

Perancangan sistem akan dijadikan acuan dalam proses implementasi sistem yang akan dibangun. Langkah kerja sistem dimulai dari pengguna memberikan

input berupa alergi makanan, jenis ADHD, dan jumlah hari yang diinginkan lalu pemrosesan optimasi menggunakan algoritme genetika, dan diakhiri dengan *output* dari sistem yang berisi rekomendasi susunan bahan makanan dan kandungan gizinya perhari sesuai jumlah hari yang dimasukkan.

3.5 Implementasi Sistem

Implementasi akan dilakukan dengan menggunakan bahasa pemrograman Javascript dengan tool Atom.

3.6 Pengujian dan Analisis Sistem

Pengujian sistem dilakukan untuk memilih parameter algoritme genetika yang terbaik sehingga dapat memberikan rekomendasi susunan bahan makanan yang optimum.

Analisis sistem dilakukan untuk mengetahui kinerja *software* yang telah dikembangkan berdasarkan hasil dari pengujian sistem. Hal ini dinilai dari pengaruh parameter-parameter yang digunakan dalam memberikan hasil atau solusi.

3.7 Pengambilan Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah tahapan perancangan, pengujian, dan analisis sistem sudah diselesaikan. Kesimpulan diambil dari hasil pengujian yang sudah dilakukan. Pada bagian akhir penulis memberikan saran yang dimaksudkan untuk dasar pertimbangan dalam pengembangan lanjutan pada sistem yang telah dirancang.

BAB 4 PERANCANGAN

Pada bab ini akan dijelaskan mengenai formulasi permasalahan, siklus Algoritme Genetika, perhitungan manual, serta perancangan antar muka.

4.1 Formulasi Permasalahan

Untuk bisa menyelesaikan permasalahan menggunakan algoritme genetika, maka diperlukan representasi dari penyelesaian asli masalah. Representasi kromosom adalah sebuah proses merubah penyelesaian asli masalah menjadi kode gen. Pengkodean kromosom yang diperlukan pada penelitian ini adalah bahan makanan yang dibutuhkan sesuai jumlah hari yang diinginkan.

4.1.1 Deskripsi Masalah

Berdasarkan beberapa penelitian yang ada, pemberian makanan dapat mempengaruhi perilaku dari anak penderita ADHD. Pemberian makanan yang sehat pada anak dapat mengurangi gejala dari ADHD. Proses penentuan susunan bahan makanan untuk anak penderita ADHD dilakukan secara manual oleh ahli gizi ataupun terapis. Ahli gizi menentukan makanan apa saja yang sebaiknya dikonsumsi anak penderita ADHD berdasarkan perhitungan nilai-nilai gizi yang terkandung dalam makanan, sedangkan untuk terapis pada umumnya menentukan hanya berdasarkan makanan apa saja yang sebaiknya dikonsumsi anak penderita ADHD berdasarkan jenis makanan tanpa adanya perhitungan nilai gizinya. Selain itu, ahli gizi ataupun terapis tidak mempertimbangkan harga dalam menentukan susunan bahan makanannya. Oleh karena itu, penulis mencoba menerapkan algoritme genetika dalam optimasi susunan bahan makanan bagi anak penderita ADHD serta memasukkan harga makanan dalam perhitungannya.

4.1.2 Daftar Bahan Makanan

Karbohidrat, protein, dan lemak merupakan zat gizi utama yang dibutuhkan oleh tubuh manusia. Kandungan karbohidrat, protein, dan lemak dalam bahan makanan dijabarkan dalam data daftar bahan makanan. Makanan yang diberikan kepada anak dalam sehari dibagi menjadi makan pagi, makan siang, dan makan malam. Variasi makanan tidak dibatasi, sehingga memungkinkan pemberian bahan makanan yang sama dalam sehari atau lebih.

4.1.3 Perhitungan Kebutuhan Gizi

Perhitungan kebutuhan gizi pada anak penderita ADHD ditentukan berdasarkan usia, jenis kelamin, berat badan (BB), tinggi badan (TB), dan aktivitas anak. Usia diperlukan untuk menghitung berat badan ideal (BBI) anak. Perhitungan kebutuhan energi anak dibedakan berdasarkan jenis kelaminnya. Kebutuhan energi anak kemudian akan dibagi menjadi tiga, yaitu kebutuhan karbohidrat, kebutuhan protein, dan kebutuhan lemak.

4.1.3.1 Data Anak

Berikut ini contoh dari data anak yang akan dijelaskan pada Tabel 4.1.

Tabel 4.1 Data Anak

No	Nama	Jenis Kelamin	Usia	Berat Badan	Tinggi Badan	Aktivitas
1	Anak	Laki-laki	6	21	118	Sedang

4.1.3.2 Perhitungan Berat Badan Ideal (BBI)

Perhitungan nilai berat badan ideal diambil dari data anak yang sudah dimasukkan menggunakan Persamaan 2.1. Data yang diperlukan untuk melakukan perhitungan BBI adalah usia anak.

- Contoh perhitungan BBI:

$$BBI_{\text{anak}} = (6 * 2) + 8 = 20 \text{ kg}$$

4.1.3.3 Perhitungan Kebutuhan Energi

Perhitungan kebutuhan energi didapatkan dengan menggunakan Persamaan 2.2. Data yang dibutuhkan meliputi berat badan ideal, tinggi badan, usia, dan faktor aktivitas.

- Contoh perhitungan kebutuhan energi:

$$\begin{aligned} \text{Energi}_{\text{anak}} &= (65 + (13,7 * 20) + (5 * 118) + (6,8 * 6)) * 1.76 \\ &= 1563,232 \text{ kkal} \end{aligned}$$

4.1.3.4 Perhitungan Kebutuhan Karbohidrat, Protein, dan Lemak perhari

Setelah didapatkan kebutuhan energi, maka selanjutnya dilakukan perhitungan kebutuhan karbohidrat, protein, dan lemak menggunakan Persamaan 2.3, 2.4, dan 2.5. Setelah dilakukan perhitungan maka didapatkan kebutuhan karbohidrat, protein, dan lemak yang akan ditunjukkan pada Tabel 4.2.

- Contoh perhitungan kebutuhan karbohidrat, protein, dan lemak:

$$KK_{\text{anak}} = \frac{1563.232 - (15\% * 1563.232 + 30\% * 1563.232)}{4} = 214.944 \text{ gram}$$

$$KP_{\text{anak}} = \frac{15\% * 1563.232}{4} = 58.621 \text{ gram}$$

$$KL_{\text{anak}} = \frac{30\% * 1563.232}{9} = 52.108 \text{ gram}$$

Tabel 4.2 Kebutuhan Zat Gizi

Nama	Kebutuhan Zat Gizi		
	Karbohidrat (gram)	Protein (gram)	Lemak (gram)
Anak	214.944	58.621	52.108

4.1.3.5 Penentuan Porsi sehari

Penentuan porsi makanan sehari ditentukan berdasarkan usia dan jenis kelamin. Pada penelitian ini diberi batasan yaitu anak dalam rentang usia satu sampai enam tahun. Anjuran porsi makanan dapat dilihat pada Tabel 4.3 yang berisi karbohidrat (makanan pokok), protein hewani (lauk hewani), protein nabati (lauk nabati), dan lemak. Pada Tabel 4.4 ditunjukkan porsi yang dibutuhkan oleh anak.

Tabel 4.3 Anjuran Porsi Sehari

Type	Usia		KH		PH		PN		Lemak	
	Batas Bawah	Batas Atas	Pria	Wanita	Pria	Wanita	Pria	Wanita	Pria	Wanita
1	1	3	3	3	1	1	1	1	3	3
2	4	6	4	4	2	2	2	2	4	4

Bisa dilihat dari Tabel 4.3 bahwa terdapat perbedaan kebutuhan porsi pada rentang usia anak satu hingga tiga tahun dan usia empat sampai enam tahun.

Tabel 4.4 Porsi Kebutuhan Gizi

Nama	KH	PH	PN	Lemak
Anak	4	2	2	4

4.2 Tahap-tahap Penyelesaian Masalah menggunakan Algoritme Genetika

Penyelesaian optimasi susunan bahan makanan untuk anak penderita ADHD menggunakan algoritme genetika seperti yang telah dijelaskan sebelumnya. Setelah mendapatkan nilai dari penghitungan kebutuhan gizi anak, variasi makanan yang akan dioptimalkan akan diproses menggunakan algoritme genetika.

Beberapa hal yang dibutuhkan dalam menyelesaikan masalah menggunakan algoritme genetika, yaitu:

Banyak hari	: 2 hari
Jenis bahan makanan	: Karbohidrat (KH), Protein Hewani (PH), Protein, Nabati (PN), Lemak (L)
Jumlah populasi (popsize)	: 3
Crossover rate (Cr)	: 0.8
Mutation rate (Mr)	: 0.6
Jumlah generasi	: 1
Angka permutasi	: [1 – 55] (indeks bahan makanan)

4.2.1 Representasi Kromosom

Dalam penelitian ini representasi kromosom yang digunakan adalah representasi permutasi dengan nilai gen bilangan permutasi pada interval 1 sampai 55 [1-55] yang mewakili indeks bahan makanan. Panjang kromosom ditentukan berdasarkan jumlah hari yang dimasukkan. Dalam sehari dibagi menjadi tiga kali waktu makan, yaitu makan pagi, siang, dan sore. Pada satu waktu makan terdapat empat asupan gizi yang diperlukan yang berupa karbohidrat (KH), protein hewani (PH), protein nabati (PN), dan lemak (L). Panjang kromosom sehari berjumlah 12 gen. Pada permasalahan ini dipilih sebanyak dua hari sehingga panjang kromosomnya berjumlah 24 gen. Pada Tabel 4.5 ditunjukkan penjelasan kromosom optimasi susunan bahan makanan selama 2 hari.

Tabel 4.5 Penjelasan Kromosom

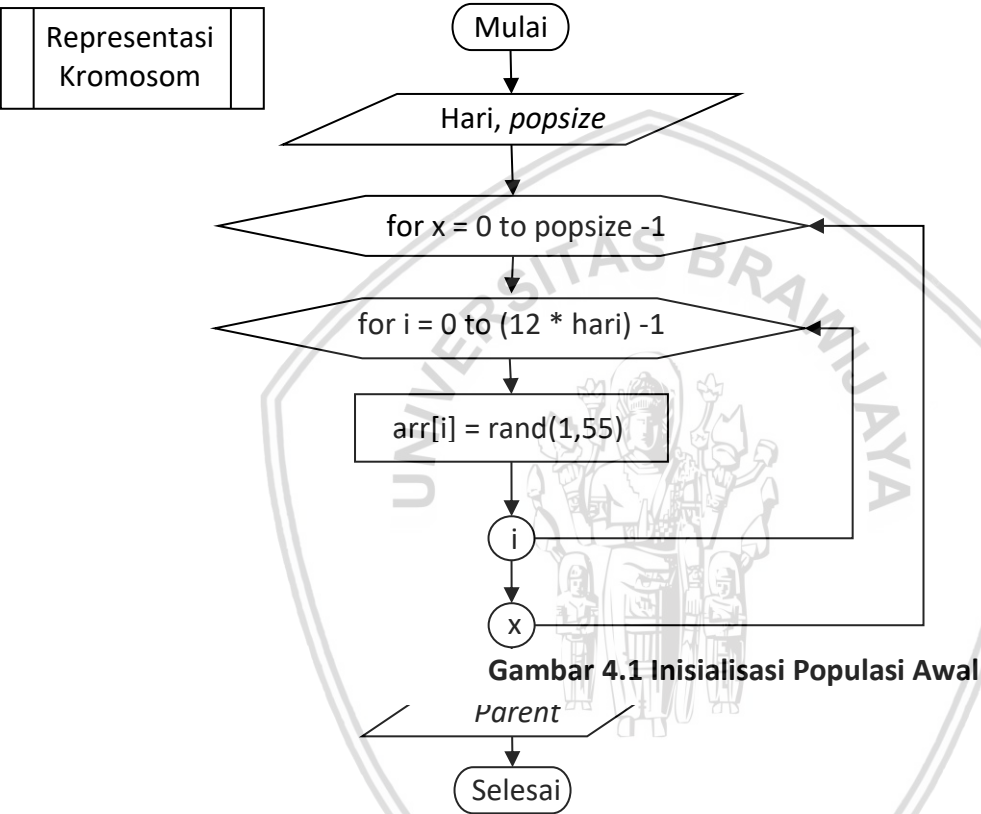
Hari 1											
Pagi				Siang				Malam			
KH	PH	PN	L	KH	PH	PN	L	KH	PH	PN	L
1	2	3	4	5	6	7	8	9	10	11	12
Hari 2											
Pagi				Siang				Malam			
KH	PH	PN	L	KH	PH	PN	L	KH	PH	PN	L
13	14	15	16	17	18	19	20	21	22	23	24

Keterangan:

KH	: Karbohidrat
PH	: Protein Hewani
PN	: Protein Nabati
L	: Lemak

4.2.2 Inisialisasi Populasi Awal

Untuk menyelesaikan permasalahan menggunakan algoritme genetika maka perlu dilakukan inisialisasi populasi awal. Nilai dari kromosom pada tiap individu didapatkan dengan cara mengambil nilai random dengan interval 1 sampai 55. Pada Gambar 4.1 yang merupakan *flowchart* proses inisialisasi dijelaskan tahapan-tahapannya. Pada Tabel 4.6 ditunjukkan contoh dari pembangkitan individu secara random.



Pada contoh telah ditentukan jumlah populasi awal berjumlah 3 individu dengan panjang gen 24 pada tiap-tiap individu. Hasil dari pembangkitan individu baru berupa P1, P2, dan P3.

Tabel 4.6 Populasi Awal

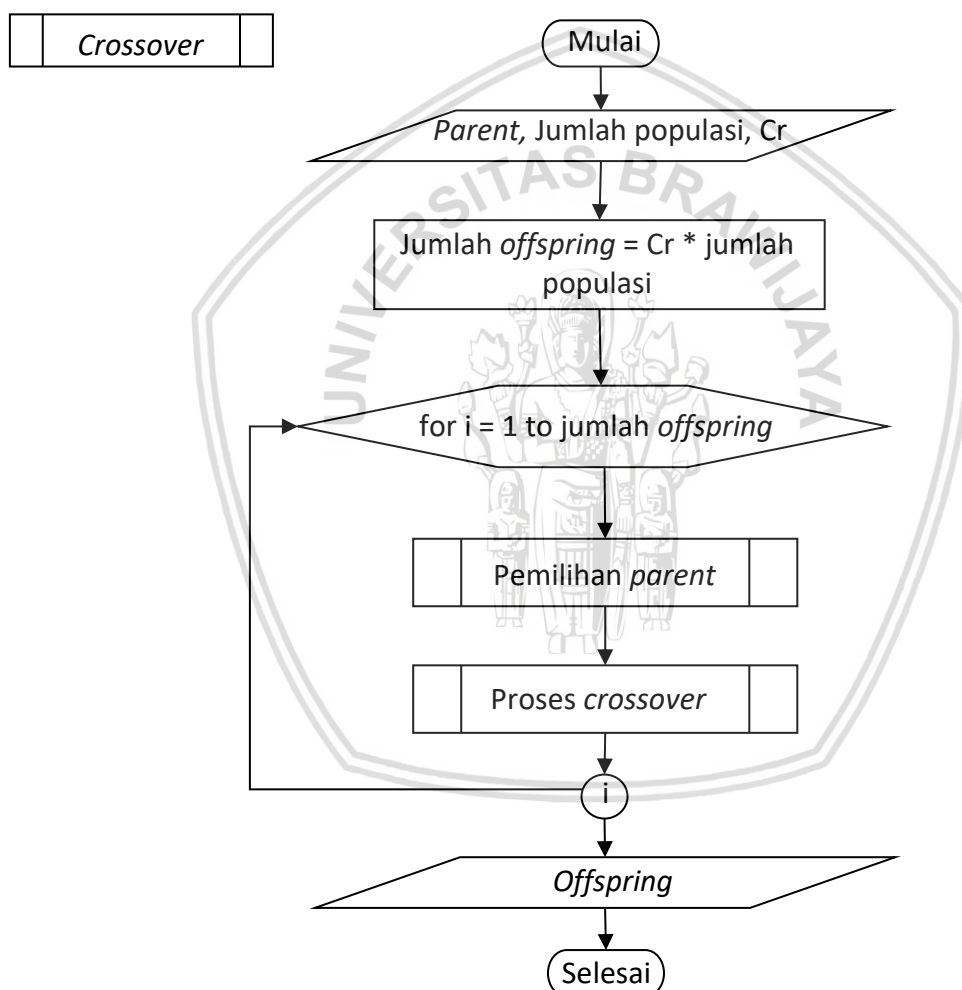
Individu	Kromosom											
P1	15	40	38	9	54	42	23	16	27	28	51	32
	24	11	35	49	29	5	19	38	16	52	40	54
P2	53	45	23	19	29	24	35	40	49	38	39	52
	2	9	51	28	15	25	17	31	26	29	34	20

P3	38	51	21	27	49	14	20	18	30	32	46	4
	2	23	26	7	19	38	18	11	32	41	42	19

4.2.3 Reproduksi

Dalam algoritme genetika terdapat dua jenis reproduksi, yaitu *crossover* dan *mutation*. Reproduksi diperlukan untuk mendapatkan individu baru yang disebut dengan *offspring*.

4.2.3.1 Crossover



Gambar 4.2 Flowchart Crossover

Tahapan dari proses *crossover* ditunjukkan pada Gambar 4.2. Sebelum melakukan *crossover*, hal pertama yang dilakukan adalah menentukan jumlah *offspring*. Hal ini didapatkan dengan mengalikan *Cr* dengan jumlah populasi awal. Pada kasus ini ditentukan *crossover rate* (*Cr*) bernilai 0,8.

- Contoh perhitungan manual:

$$offspring = 3 * 0,8 = 2,4 \approx 2$$

Dari perhitungan diatas didapatkan jumlah *offspring* sebanyak 2. Selanjutnya dipilih dua induk secara acak yang kemudian akan dilakukan *crossover*. Pada permasalahan ini, *crossover* yang digunakan adalah *one-cut point crossover*. Dalam sekali proses *crossover* menghasilkan maksimal dua individu baru. Pada Tabel 4.7 menunjukkan proses *crossover* antara P1 dan P2 yang menghasilkan *offspring* C1 dan C2.

Tabel 4.7 Proses Crossover

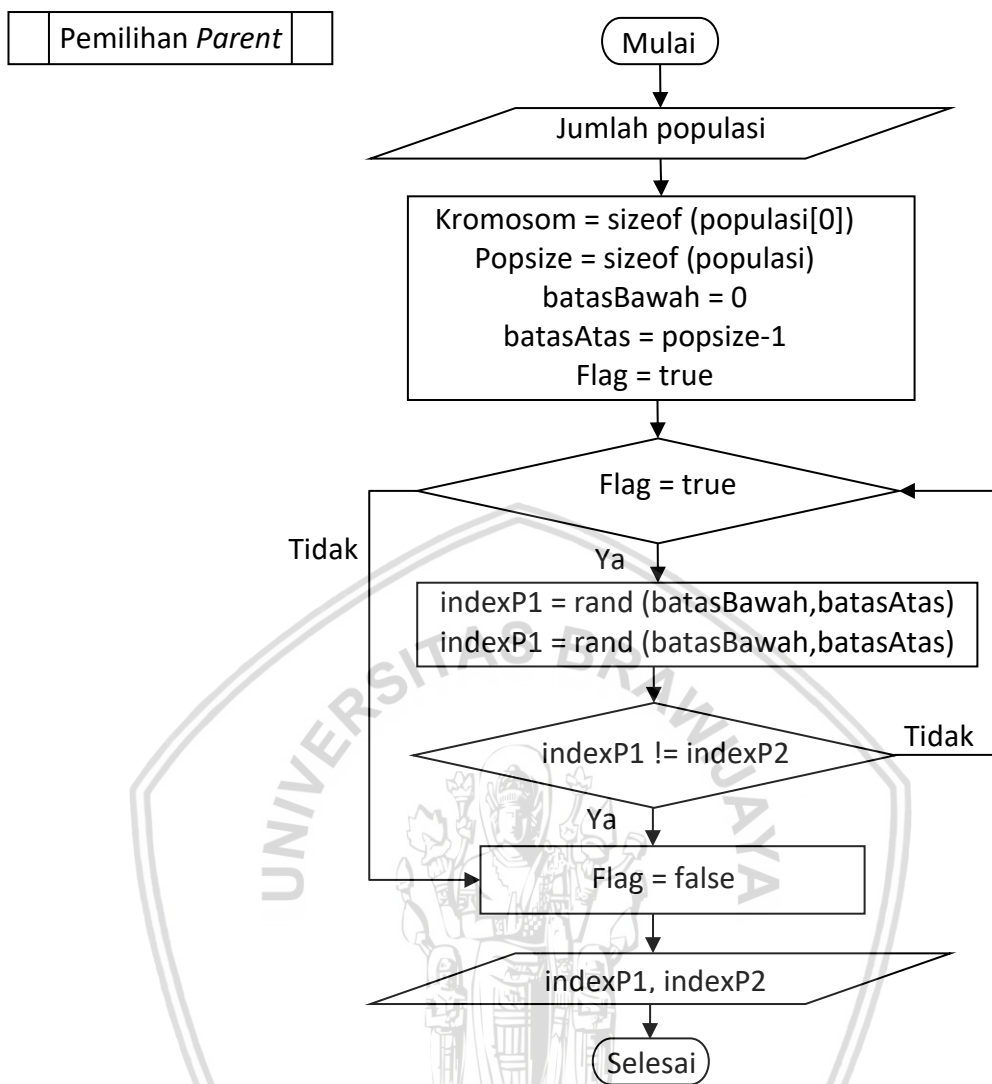
Individu	Kromosom											
P1	15	40	38	9	54	42	23	16	27	28	51	32
	24	11	35	49	29	5	19	38	16	52	40	54
P2	53	45	23	19	29	24	35	40	49	38	39	52
	2	9	51	28	15	25	17	31	26	29	34	20
C1	15	40	38	9	54	42	23	16	27	28	51	32
	2	9	51	28	15	25	17	31	26	29	34	20
C2	53	45	23	19	29	24	35	40	49	38	39	52
	24	11	35	49	29	5	19	38	16	52	40	54

Pada Tabel 4.7 bisa dilihat proses dari *crossover*. Pertama melakukan pembagian kromosom menjadi dua bagian pada P1 dan P2. Selanjutnya melakukan penggabungan pada potongan pertama dari kromosom P1 dan potongan kedua dari kromosom P2 yang menghasilkan individu baru C1. Hal ini dilakukan juga pada potongan pertama dari kromosom P2 dan potongan kedua dari kromosom P1 yang menghasilkan individu baru C2.

1. Proses Pemilihan Parent

Dalam melakukan *crossover* diperlukan dua individu sebagai *parent* yang akan menghasilkan maksimal dua individu baru sebagai *offspring*. Calon *parent* diambil dari individu pada populasi awal. Pada permasalahan ini dibutuhkan dua *offspring*, sehingga proses *crossover* yang diperlukan berjumlah satu kali.

Pada Gambar 4.3 dijelaskan tahap-tahap yang dilakukan untuk memilih *parent* yang selanjutnya akan digunakan dalam *crossover*. Pemilihan *parent* dilakukan secara *random* berdasarkan indeksinya. Pemilihan dilakukan terus menerus hingga didapatkan *parent* 1 dan *parent* 2 yang didapatkan memiliki indeks yang berbeda.

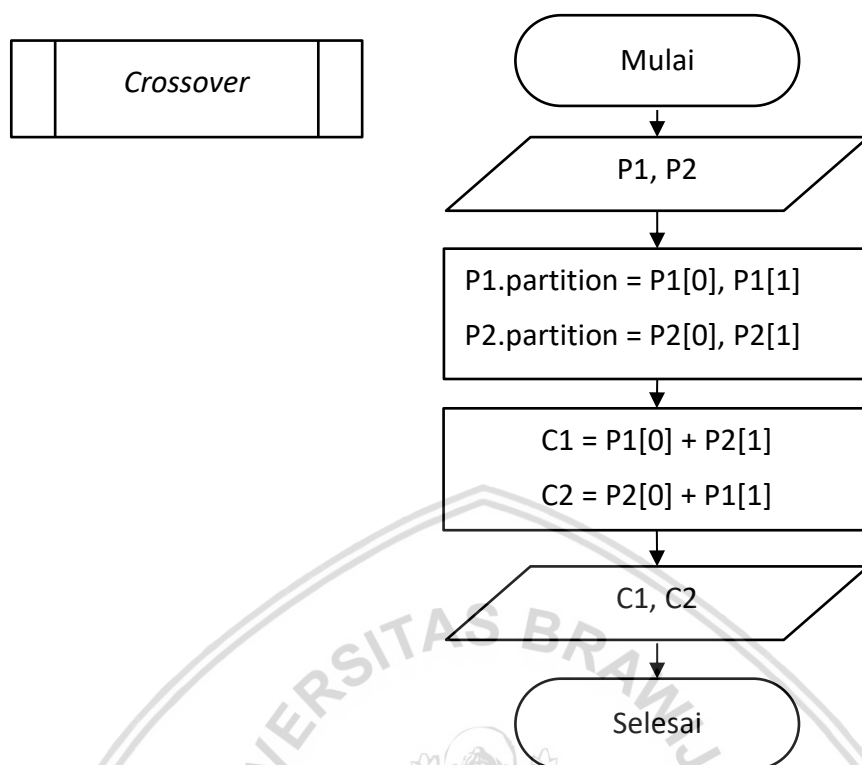


Gambar 4.3 Proses Pemilihan *Parent*

2. Proses *One-cut Point Crossover*

Tahapan-tahapan dalam proses *crossover* menurut Gambar 4.4 adalah sebagai berikut:

1. Melakukan inisialisasi P1 dan P2
2. Menentukan titik potong dari P1 dan P2 yang diperoleh dari panjang kromosom dibagi 2.
3. Menggabungkan potongan pertama dari P1 dengan potongan kedua dari P2 dan potongan pertama dari P2 dengan potongan kedua dari P1
4. Memperoleh nilai C1 dan C2.



Gambar 4.4 Flowchart One-cut Point Crossover

4.2.3.2 Mutation

Tahap selanjutnya dari proses reproduksi adalah melakukan proses mutasi. Proses mutasi dimulai dengan menentukan jumlah *offspring* yang harus dihasilkan. Jumlah *offspring* didapatkan dengan mengalikan Mr dengan jumlah populasi awal. Pada contoh kasus ini ditentukan *mutation rate* (Mr) 0,6.

- Contoh perhitungan manual

$$\text{offspring} = 3 \times 0,6 = 1,8 \approx 2$$

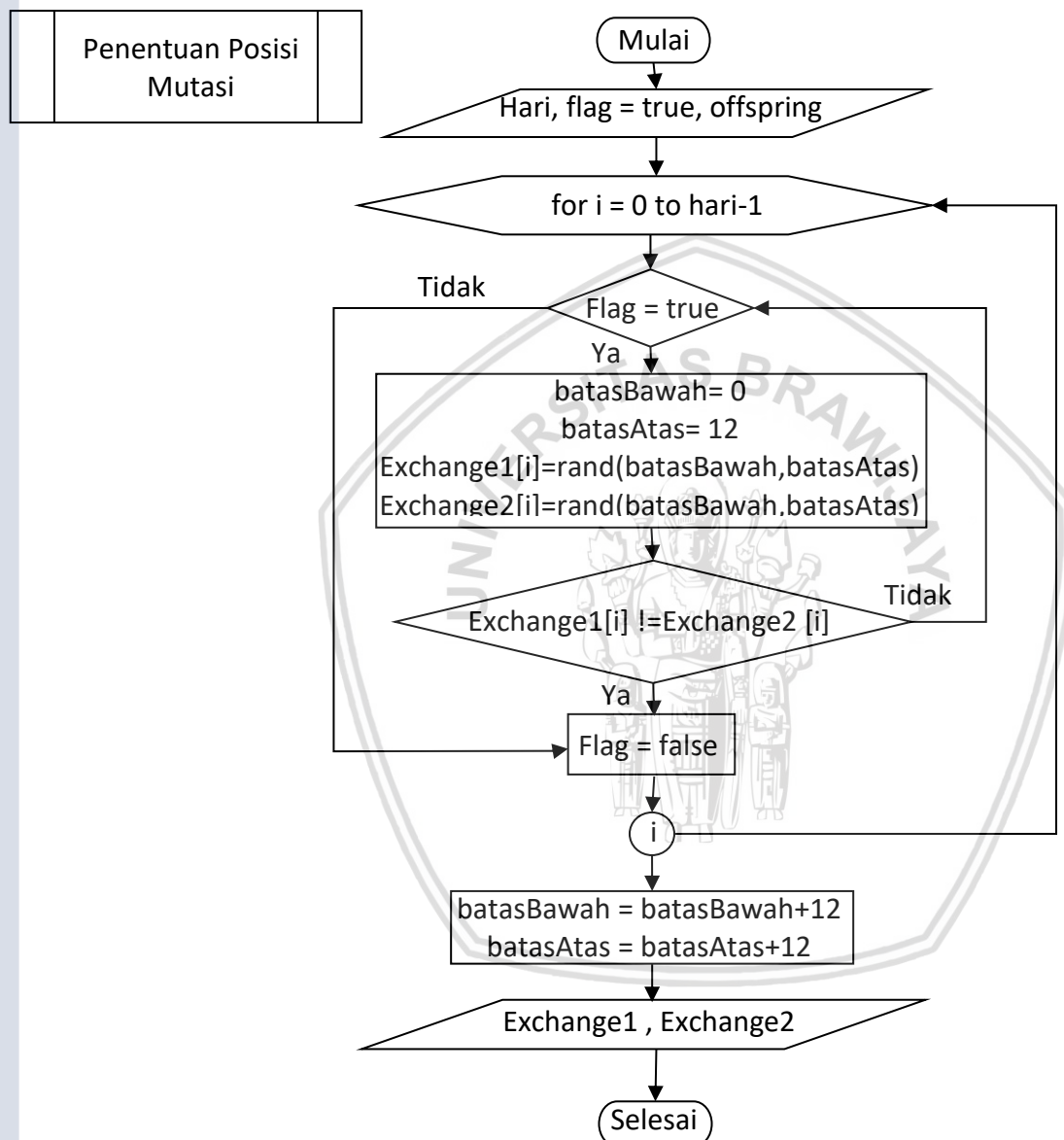
Pada perhitungan diatas didapatkan *offspring* yang harus dihasilkan berjumlah 2. Selanjutnya melakukan proses pemilihan *parent* secara *random*. Setelah itu menentukan posisi dari kromosom yang akan dilakukan proses mutasi.

Tabel 4.8 Exchange Mutation

Individu	Kromosom											
P1	15	40	38	9	54	42	23	16	27	28	51	32
	24	11	35	49	29	5	19	38	16	52	40	54
C3	15	40	27	9	54	42	23	16	38	28	51	32
	24	16	35	49	29	5	19	38	11	52	40	54
P3	38	51	21	27	49	14	20	18	30	32	46	4
	2	23	26	7	19	38	18	11	32	41	42	19
C4	38	51	21	27	32	14	20	18	30	49	46	4
	18	23	26	7	19	38	2	11	32	41	42	19

Pada Tabel 4.8 ditunjukkan bahwa proses mutasi dimulai dengan membagi kromosom menjadi 2 segmen. Pembagian segmen didapatkan berdasarkan jumlah hari yang diinginkan, contoh pada kasus ini 2 hari. Penentuan posisi mutasi pada tiap segmen didapatkan secara *random*.

1. Proses Penentuan Posisi Gen Mutasi



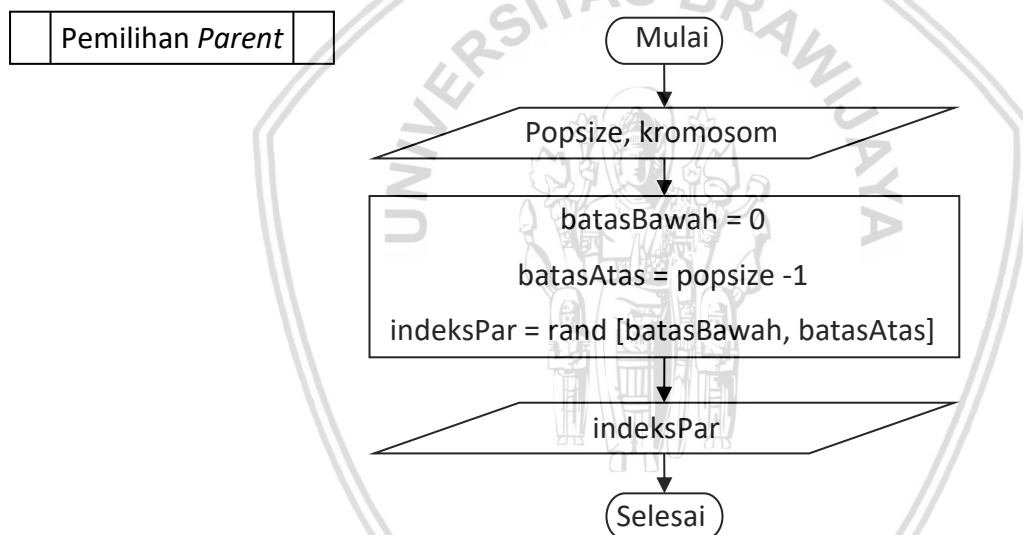
Gambar 4.5 Flowchart Penentuan Posisi Gen Mutasi

Beberapa tahapan dalam penentuan posisi gen mutasi berdasarkan Gambar 4.5 adalah sebagai berikut:

1. Melakukan inisialisasi pada variable *batasAtas* dan *batasBawah* sebagai batasan untuk angka *random*.

2. Melakukan perulangan dari segmen pertama hingga segmen terakhir. Pembagian segmen ditetapkan tiap 12 gen yang merupakan pembagian bahan makanan sehari.
3. Inisialisasi variable flag yang digunakan dalam perulangan *while*.
4. Menentukan posisi gen dengan melakukan *random* pada indeks gen.
5. Posisi gen yang akan ditukar harus berbeda. Ketika posisinya sama maka kembali ke tahap 4.
6. Setelah mendapatkan posisi gen yang berbeda maka merubah nilai variable flag menjadi *false*.
7. Penambahan nilai batasAtas dan batasBawah dengan nilai 12 untuk pencarian posisi mutasi pada segmen berikutnya.
8. Menghasilkan posisi exchange1 dan exchange2

2. Proses Pemilihan Parent

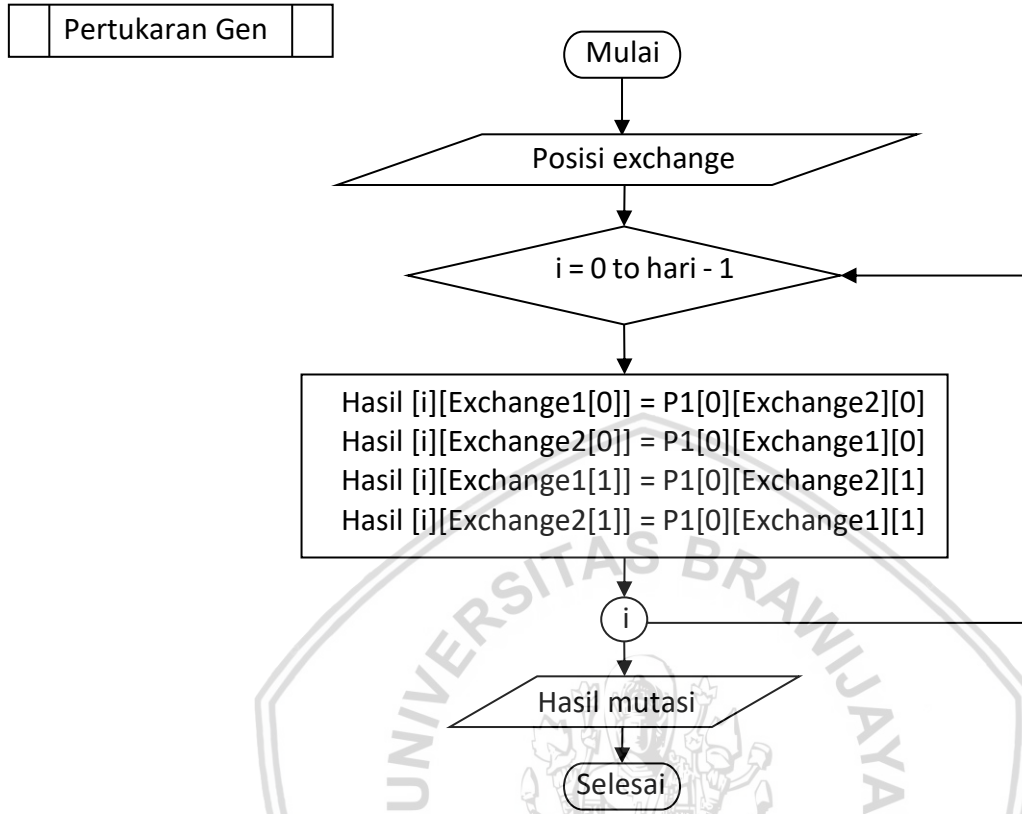


Gambar 4.6 Flowchart Pemilihan Parent

Tahap-tahap untuk memilih *parent* berdasarkan Gambar 4.6 adalah sebagai berikut:

1. Melakukan inisialisasi batasAtas, batasBawah, dan indeksPar.
2. Mendapatkan nilai *random* sebagai indeksPar.
3. Mendapatkan *parent* sesuai dengan indeksPar.

3. Proses Pertukaran Gen



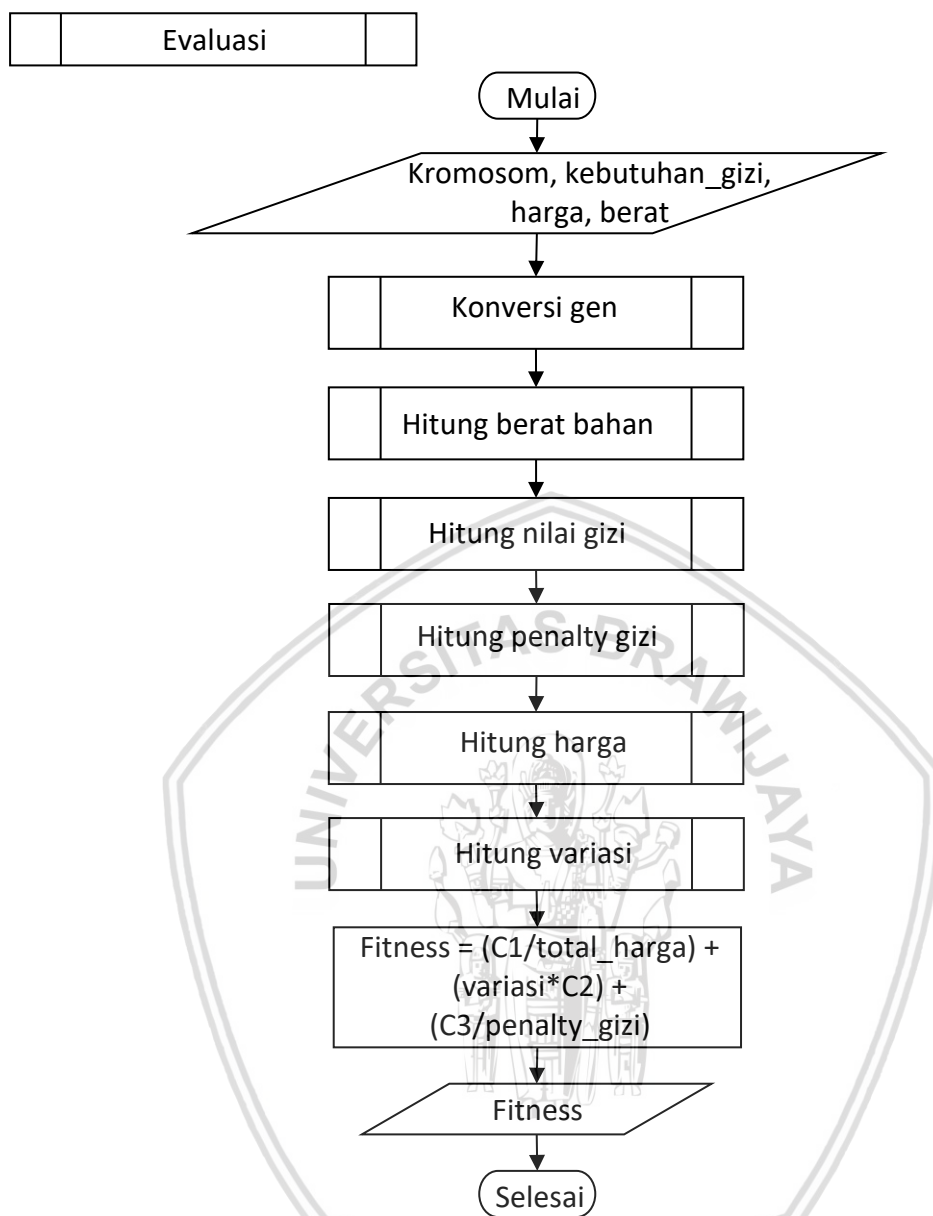
Gambar 4.7 Flowchart Pertukaran Gen

Tahapan yang ditunjukkan pada Gambar 4.7 adalah sebagai berikut:

1. Inisialisasi posisi *exchange*.
2. Melakukan perulangan sebanyak jumlah hari.
3. Menukar nilai dari gen sesuai dengan posisi *exchange* yang telah ditentukan pada proses sebelumnya.
4. Melakukan perulangan pada tahap 3 hingga batas perulangan.
5. Mendapatkan hasil dari proses mutasi.

4.2.4 Evaluasi

Nilai *fitness* digunakan untuk mengetahui seberapa optimal solusi yang didapat dari suatu kromosom. Pada Gambar 4.8 ditunjukkan tahap dalam perhitungan nilai *fitness*.

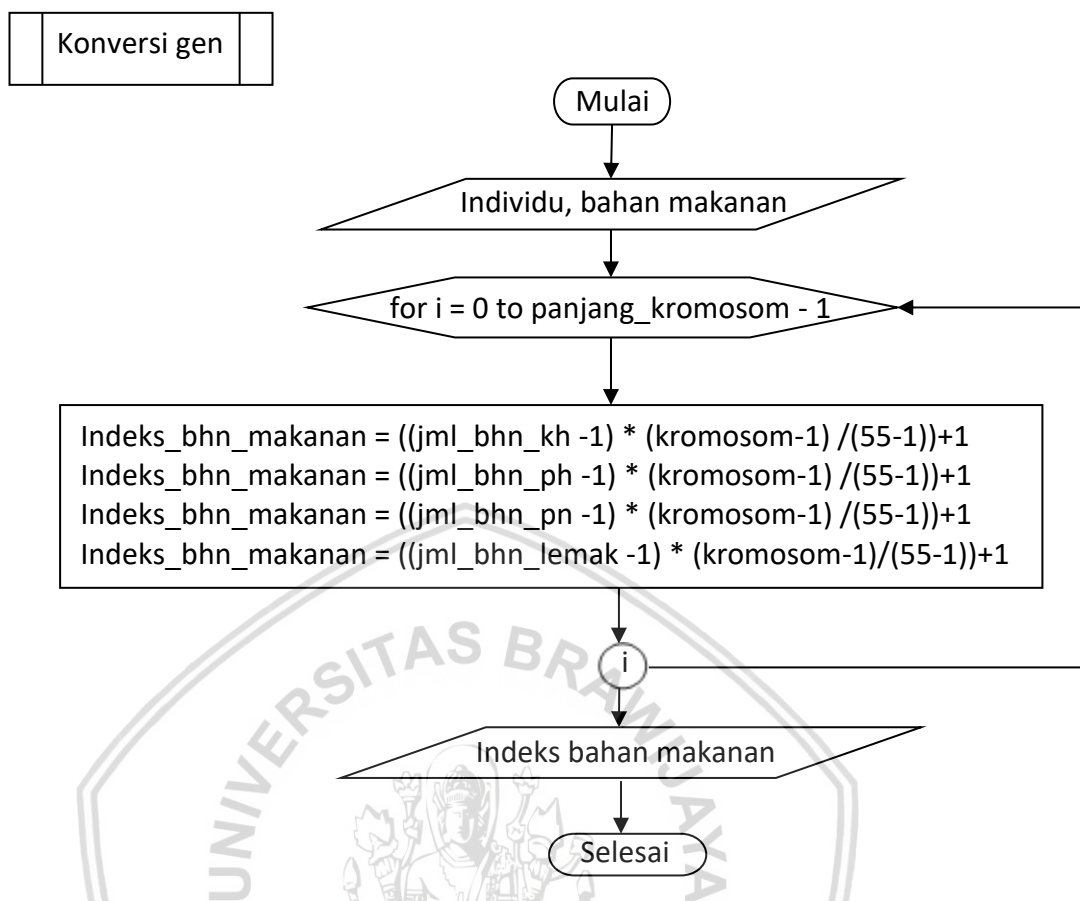


Gambar 4.8 Flowchart Evaluasi

Berdasarkan Gambar 4.8 untuk mendapatkan nilai *fitness* melalui tahap-tahap sebagai berikut:

1. Konversi Nilai Gen Menjadi Indeks Bahan Makanan

Hal pertama yang harus dilakukan dalam melakukan evaluasi adalah melakukan konversi nilai gen menjadi indeks bahan makanan. Tahap-tahap dalam melakukan konversi nilai gen menjadi indeks bahan makanan ditunjukkan pada Gambar 4.9.



Gambar 4.9 Flowchart Konversi Gen

Pada Gambar 4.9 ditunjukkan *flowchart* dari proses konversi gen menjadi indeks bahan makanan. Nilai gen adalah nilai *random* pada interval 1 sampai 55. Nilai ini kemudian akan dikonversi menjadi indeks bahan makanan. Hasil dari proses konversi kemudian dibulatkan ke angka penting sesuai dengan jumlah data pada jenis bahan makanan.

Berikut ini adalah contoh dari perhitungan manual dari konversi gen menjadi indeks bahan makanan:

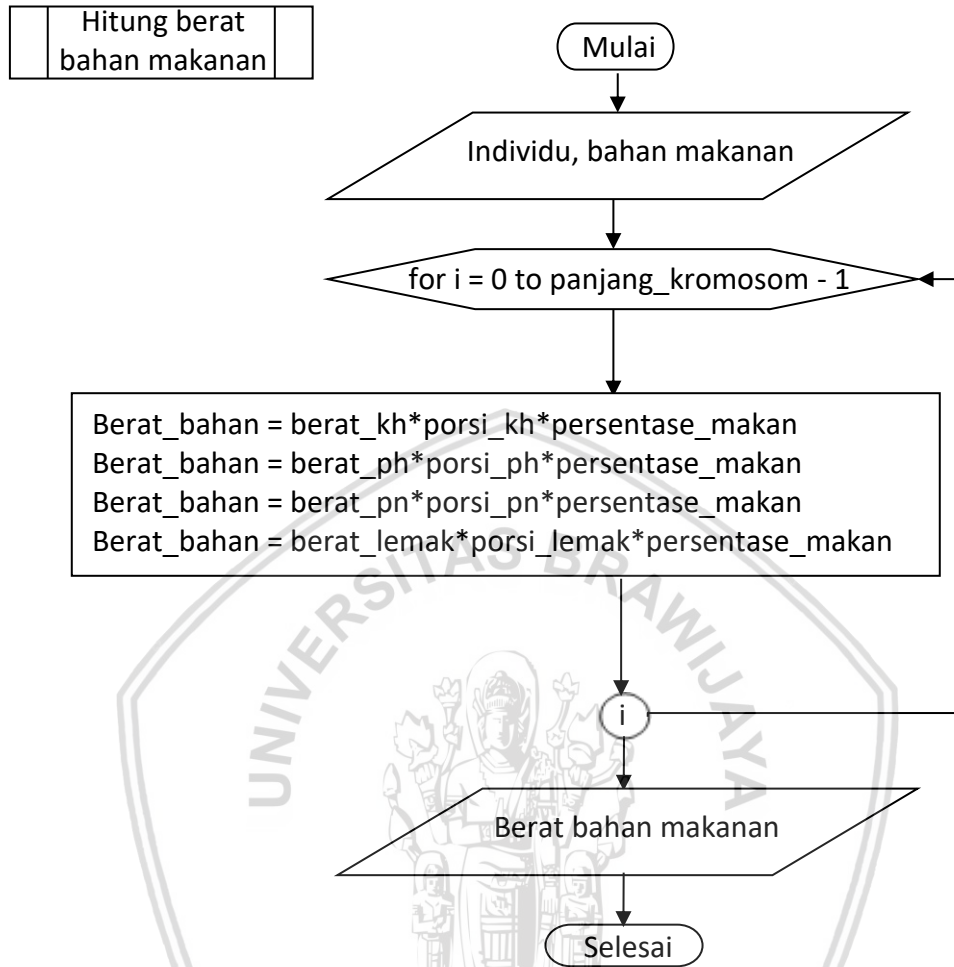
- $Gen\ 1 = \left((17 - 1) * \frac{15-1}{55-1} \right) + 1 = 5,1481 \approx 5$
- $Gen\ 2 = \left((38 - 1) * \frac{40-1}{55-1} \right) + 1 = 28,722 \approx 29$
- $Gen\ 3 = \left((12 - 1) * \frac{38-1}{55-1} \right) + 1 = 9,2222 \approx 9$
- $Gen\ 4 = \left((12 - 1) * \frac{9-1}{55-1} \right) + 1 = 2,6296 \approx 3$

Tabel 4.9 merupakan hasil konversi dari nilai gen pada kromosom P1 menjadi indeks bahan makanan beserta nama bahan makanan sesuai dengan indeks hasil konversi nilai gen.

Tabel 4.9 Hasil Konversi Gen

Hari ke-	Waktu Makan	Jenis Bahan Makanan	Gen	Index Bahan Makanan	Nama Makanan
1	Pagi	KH	15	5	Krekers
		PH	40	29	Otak
		PN	38	9	Oncom
		L	9	3	Mayonnaise
	Siang	KH	54	17	Tape Singkong
		PH	42	31	Telur bebek asin
		PN	23	6	Kacang Tanah Kupas
		L	16	4	Minyak Jagung
	Malam	KH	27	9	Nasi Beras Giling
		PH	28	21	Teri kering
		PN	51	12	Tahu
		L	32	7	Mentega
2	Pagi	KH	2	1	Bihun
		PH	9	7	Ikan asin kering
		PN	51	12	Tahu
		L	28	7	Mentega
	Siang	KH	15	5	Krekers
		PH	25	18	Putih telur ayam
		PN	17	5	Kacang Tanah
		L	31	7	Mentega
	Malam	KH	26	8	Mi Kering
		PH	29	21	Teri kering
		PN	34	8	Kembang Tahu
		L	20	5	Minyak Kacang Kedele

2. Menghitung Berat Bahan Makanan



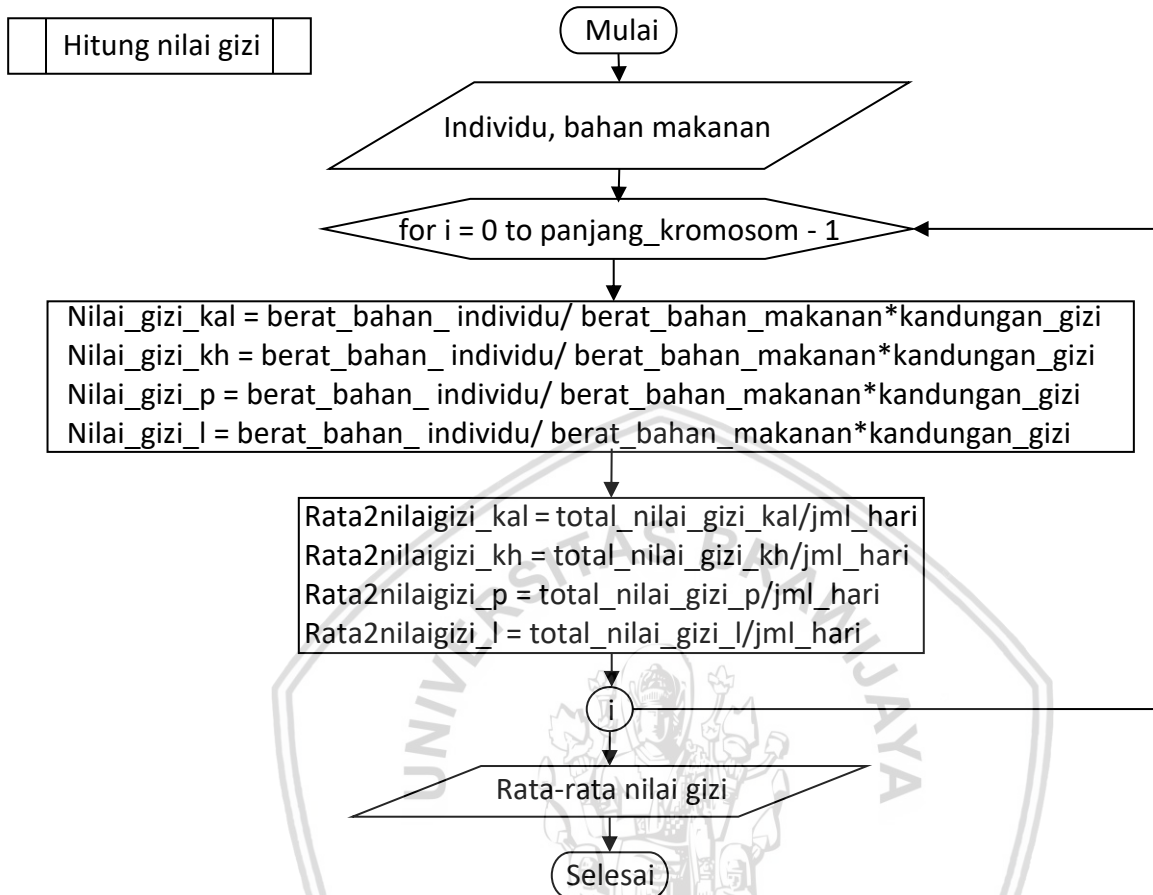
Gambar 4.10 Flowchart Hitung Berat Bahan Makanan

Tahap selanjutnya adalah menghitung berat yang dibutuhkan dari tiap-tiap bahan makanan. Untuk mendapatkan berat bahan makanan, pembagian porsi makanan pada pagi, siang, dan malam diperlukan untuk menghitungnya. Pada Gambar 4.10 ditunjukkan urutan untuk menghitung berat bahan makanan yang dibutuhkan.

Berikut ini adalah contoh dari perhitungan manual untuk gen 1 pada kromosom P1:

- $bkh_{anak} = 50 * 4 * 0,25 = 50 \text{ gram}$

3. Menghitung Nilai Gizi Setiap Bahan Makanan



Gambar 4.11 Flowchart Perhitungan Nilai Gizi

karbohidrat, protein, dan lemak yang dimiliki oleh tiap-tiap bahan makanan. Pada Gambar 4.11 ditunjukkan langkah-langkah yang harus dilakukan untuk menghitung nilai gizi tersebut.

- Contoh perhitungan manual kandungan gizi gen 1 pada kromosom P1:

$$Kal_{anak} = \frac{50}{50} * 175 = 175 \text{ kkal}$$

$$KH_{anak} = \frac{50}{50} * 40 = 40 \text{ gram}$$

$$P_{anak} = \frac{50}{50} * 4 = 4 \text{ gram}$$

$$L_{anak} = \frac{50}{50} * 0 = 0 \text{ gram}$$

Setelah didapatkan total nilai gizi kalori, karbohidrat, protein, dan lemak pada tiap bahan makanan yang dibutuhkan, selanjutnya menghitung rata-rata nilai gizi yang terpenuhi dengan melakukan pembagian dengan jumlah hari. Pada contoh masalah disini jumlah hari adalah 2 hari.

- Contoh perhitungan manual rata-rata gizi pada kromosom P1:

$$Kal_{anak} = \frac{2402,50}{2} = 1201,25 \text{ kkal}$$

$$KH_{anak} = \frac{348}{2} = 174 \text{ gram}$$

$$P_{anak} = \frac{80}{2} = 40 \text{ gram}$$

$$L_{anak} = \frac{71,6}{2} = 35,8 \text{ gram}$$

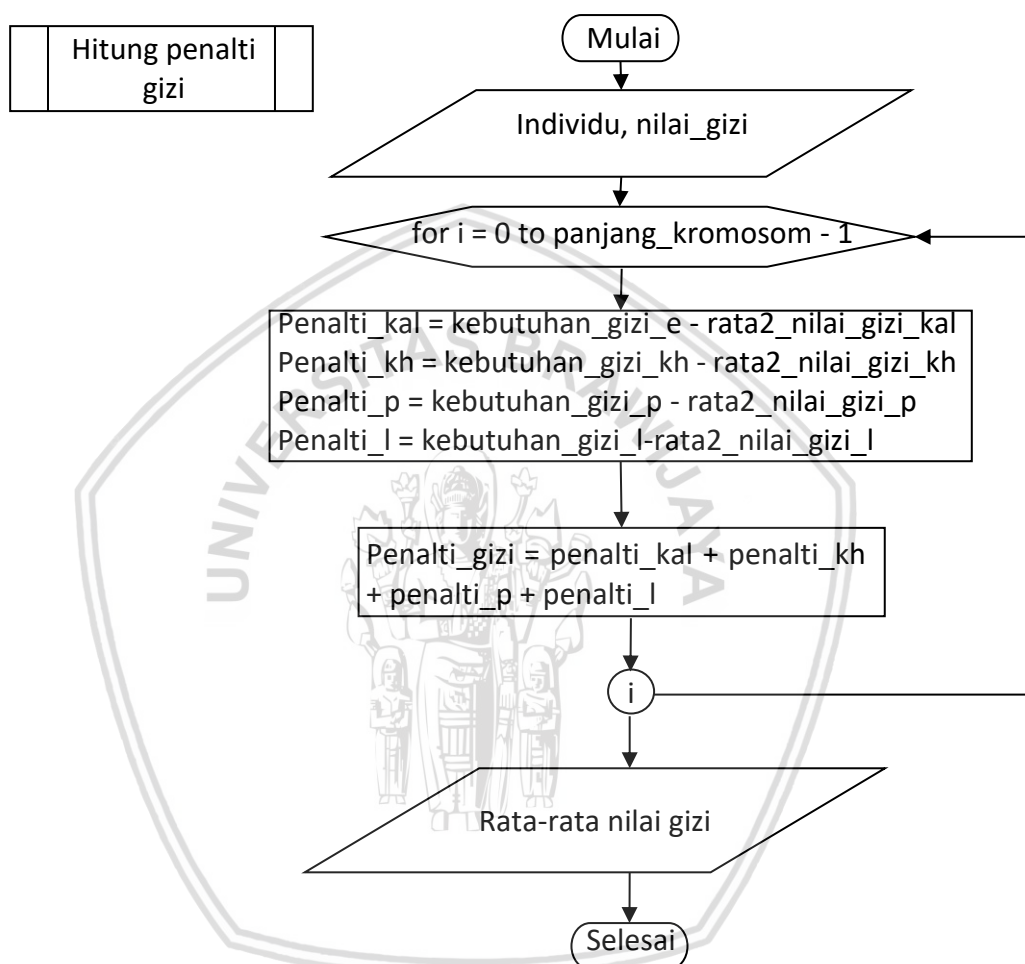
Hasil terakhir dari perhitungan berat dan nilai gizi pada kromosom P1 ditunjukkan pada Tabel 4.10 berikut ini:

Tabel 4.10 Berat dan Nilai Gizi

Indeks Bahan Makanan	Nama Makanan	Berat	Kalori	Karbohidrat	Protein	Lemak
		Anak	Anak	Anak	Anak	Anak
5	Krekers	50	175	40	4	0
29	Otak	17.5	37.5	0	3.5	2.5
9	Oncom	20	37.5	3.5	2.5	1.5
3	Mayonnaise	20	50	0	0	5
17	Tape Singkong	160	280	64	6.4	0
31	Telur bebek asin	40	60	0	5.6	4
6	Kacang Tanah Kupas	12	60	5.6	4	2.4
4	Minyak Jagung	8	80	0	0	8
9	Nasi Beras Giling	140	245	56	5.6	0
21	Teri kering	10.5	35	0	4.9	1.4
12	Tahu	77	52.5	4.9	3.5	2.1
7	Mentega	7	70	0	0	7
8	Mi Kering	50	175	40	4	0
8	Ikan kakap	17.5	25	0	3.5	1
9	Oncom	20	37.5	3.5	2.5	1.5
11	Minyak Kelapa	5	50	0	0	5
9	Nasi Beras Giling	160	280	64	6.4	0
4	Daging Ayam tanpa kulit	32	40	0	5.6	1.6
5	Kacang Tanah	12	60	5.6	4	2.4
9	Kelapa	24	80	0	0	8
5	Krekers	70	245	56	5.6	0
38	Sardencis	24.5	105	0	4.9	9.1
10	Pete Segar	38.5	52.5	4.9	3.5	2.1

12	Minyak Kelapa Sawit	7	70	0	0	7
Total			2402.50	348.00	80.00	71.60
Rata-rata			1201.25	174	40	35.8

4. Menghitung Penalti Gizi



Gambar 4.12 Flowchart Perhitungan Penalti Gizi

Pada Gambar 4.12 ditunjukkan langkah untuk menghitung penalty gizi. Penalti gizi adalah kebutuhan gizi yang tidak terpenuhi ataupun melebihi kebutuhan gizi yang seharusnya. Untuk mendapatkan nilai penalti gizi diperlukan penjumlahan dari penalti kalori, penalti karbohidrat, penalti protein, dan penalti lemak.

- Contoh perhitungan manual penalti gizi pada kromosom P1:

$$\text{penaltiKal}_{\text{anak}} = 1563,2 - 1201,2 = 362 \text{ kkal}$$

$$penaltiKH_{anak} = 214,94 - 174 = 40,944 \text{ gram}$$

$$penaltiP_{anak} = 58,621 - 40 = 18,621 \text{ gram}$$

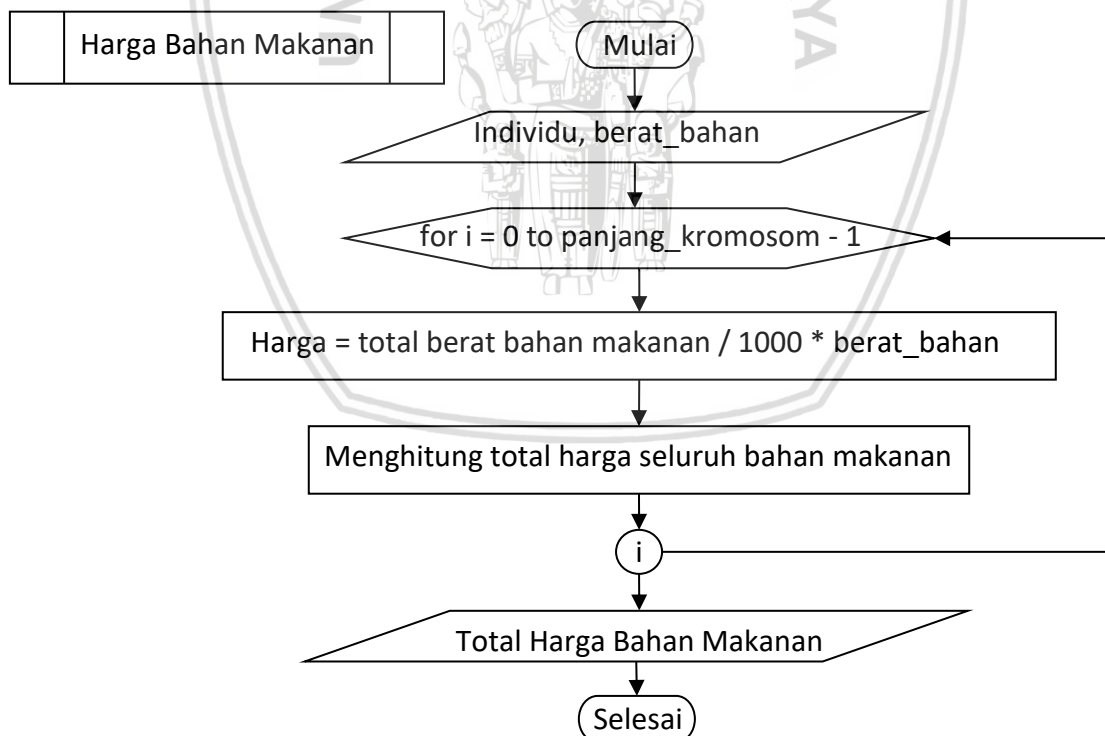
$$penaltiL_{anak} = 52,108 - 35,8 = 16,308 \text{ gram}$$

Masing-masing kromosom dihitung penalti gizinya seperti pada contoh diatas. Setelah itu menghitung total penalti gizi dengan menjumlahkan penalti kalori, penalti karbohidrat, penalti protein, dan penalti lemak. Pada Tabel 4.11 ditampilkan hasil perhitungan penalti masing-masing kebutuhan gizi beserta total penalti pada kromosom P1.

Tabel 4.11 Penalti Gizi

Jenis Gizi	Penalti
	Anak
Kalori	362
Karbohidrat	40,944
Protein	18,621
Lemak	16,308
Total penalti	437,86

5. Menghitung Harga Bahan Makan



Gambar 4.13 Flowchart Menghitung Harga Bahan Makanan

Pada Gambar 4.13 ditunjukkan langkah-langkah dalam menghitung masing-masing harga bahan makanan. Harga yang tertera pada data bahan makanan merupakan harga perkilo, sedangkan berat bahan makanan yang diperlukan

ditunjukkan dalam satuan gram. Oleh karena itu perlu dilakukan pembagian dengan 1000 untuk mendapatkan harga bahan makanan pergram.

- Contoh perhitungan manual harga bahan makanan pada gen 1 kromosom P1:

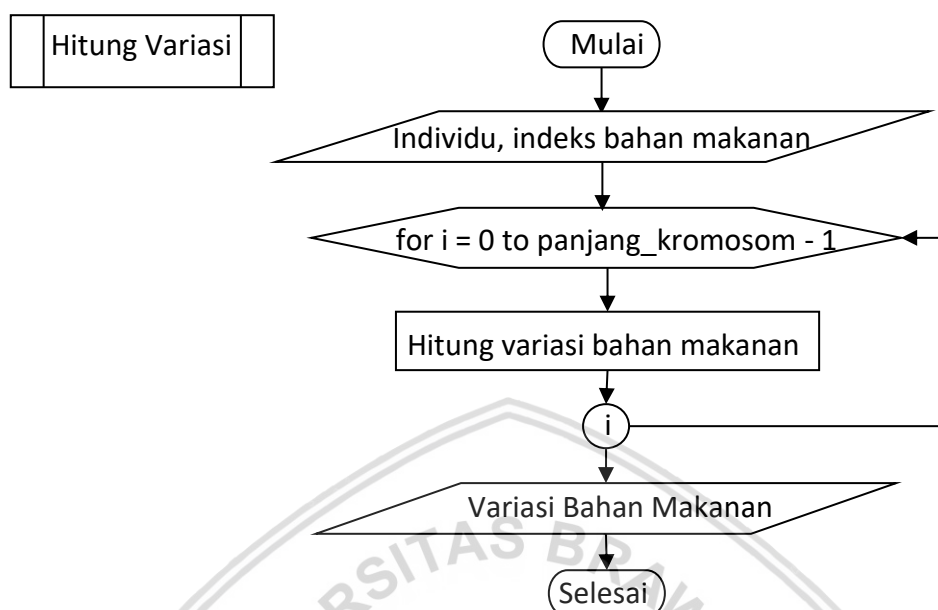
$$\text{harga} = \frac{50}{1000} * 45000 = 2250$$

Pada Tabel 4.12 menunjukkan harga masing-masing bahan makanan sesuai dengan berat bahan makanan yang dibutuhkan dan total harga dari bahan makanan yang dibutuhkan pada kromosom P1.

Tabel 4.12 Harga Bahan Makanan

Index Bahan Makanan	Nama Makanan	Berat	Harga
		Anak1	
5	Krekers	50	2250.00
29	Otak	17.5	525.00
9	Oncom	20	700.00
3	Mayonnaise	20	600.00
17	Tape Singkong	160	3200.00
31	Telur bebek asin	40	70.00
6	Kacang Tanah Kupas	12	264.00
4	Minyak Jagung	8	824.00
9	Nasi Beras Giling	140	1414.00
21	Teri kering	10.5	609.00
12	Tahu	77	770.00
7	Mentega	7	350.00
8	Mi Kering	50	1000.00
8	Ikan kakap	17.5	1400.00
9	Oncom	20	700.00
11	Minyak Kelapa	5	134.00
9	Nasi Beras Giling	160	1616.00
4	Daging Ayam tanpa kulit	32	768.00
5	Kacang Tanah	12	264.00
9	Kelapa	24	192.00
5	Krekers	70	3150.00
38	Sardencis	24.5	1036.35
10	Pete Segar	38.5	385.00
12	Minyak Kelapa Sawit	7	77.00
Total			22298.35

6. Menghitung Variasi Bahan Makanan



Gambar 4.14 Flowchart Menghitung Variasi Bahan Makanan

Pada Gambar 4.14 menunjukkan langkah-langkah dalam menghitung variasi bahan makanan. Perhitungan variasi diperlukan untuk mendapatkan keanekaragaman bahan makanan, dimana anak-anak mudah bosan jika diberikan makanan yang sama secara terus-menerus. Pada Tabel 4.13 menampilkan hasil dari perhitungan variasi pada kromosom P1. Dalam tabel ditunjukkan pada bagian karbohidrat terdapat 4 bahan makanan yang sama sehingga variasinya bernilai 2.

Tabel 4.13 Variasi Bahan Makanan

Jenis Bahan	Hari 1			Hari 2			Total
Makanan	Pagi	Siang	Malam	Pagi	Siang	Malam	
KH	5	17	9	8	9	5	2
PH	29	31	21	8	4	38	6
PN	9	6	12	9	5	10	4
L	3	4	7	11	9	12	6
Jumlah Variasi							18

7. Menghitung Nilai *Fitness*

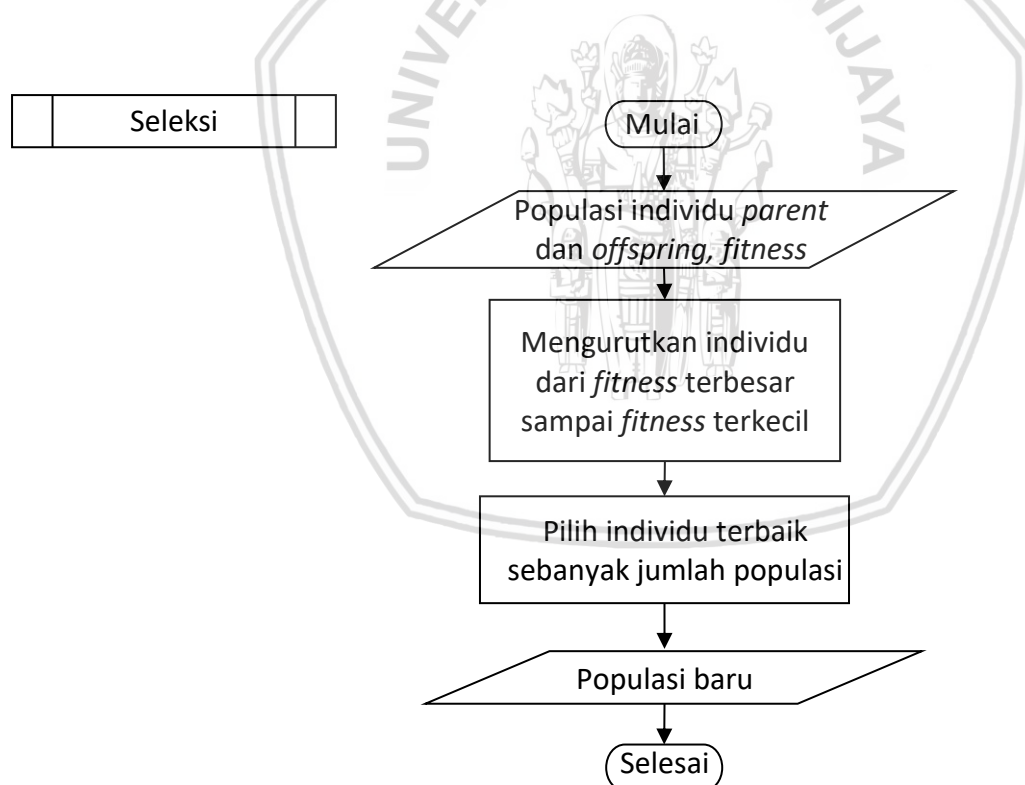
Setelah mendapatkan nilai variasi maka dilanjutkan proses perhitungan *fitness* dari setiap kromosom. Dikarenakan solusi yang dicari adalah nilai pinalti gizi dan total harga minimum maka Persamaan 2.3 digunakan, sedangkan untuk variasi dicari yang memiliki tingkat keragaman tinggi sehingga menggunakan Persamaan 2.1 untuk nilai maksimum. Nilai C yang digunakan pada pinalti gizi dan total harga bernilai 1000.

- Contoh perhitungan *fitness* untuk kromosom P1:

$$fitness = \frac{1000}{22298,5} + 18 + \frac{1000}{437,8553} = 20,3287$$

4.2.4.1 Seleksi

Seleksi digunakan untuk mendapatkan individu terbaik berdasarkan nilai *fitness*. Pada penelitian ini metode seleksi yang digunakan adalah *elitism selection*. Pada metode ini *parent* dan *offspring* dikumpulkan yang kemudian diurutkan berdasarkan nilai *fitness* dari yang paling tinggi ke yang paling rendah. Individu terbaik akan dipilih untuk melanjutkan ke generasi berikutnya sebagai *parent*. Pada Gambar 4.15 menunjukkan tahap-tahap dalam proses seleksi.



Gambar 4.15 Flowchart Proses Seleksi

1. Populasi yang terdiri dari individu *parent* dan *offspring* beserta nilai *fitness*-nya. Pada Tabel 4.14 menunjukkan populasi *parent* dan *offspring* beserta dengan nilai *fitness*-nya.

Tabel 4.14 Populasi *Parent* dan *Offspring*

Individu	Kromosom												Fitness
P1	15	40	38	9	54	42	23	16	27	28	51	32	20.3287
	24	11	35	49	29	5	19	38	16	52	40	54	
P2	53	45	23	19	29	24	35	40	49	38	39	52	18.1280
	2	9	51	28	15	25	17	31	26	29	34	20	
P3	38	51	21	27	49	14	20	18	30	32	46	4	19.2699
	2	23	26	7	19	38	18	11	32	41	42	19	
C1	15	40	38	9	54	42	23	16	27	28	51	32	17.1386
	2	9	51	28	15	25	17	31	26	29	34	20	
C2	53	45	23	19	29	24	35	40	49	38	39	52	17.3161
	24	11	35	49	29	5	19	38	16	52	40	54	
C3	15	40	27	38	9	54	42	23	16	28	51	32	18.5075
	24	16	11	35	49	29	5	19	38	52	40	54	
C4	38	51	21	27	32	49	14	20	18	30	46	4	18.4543
	18	2	23	26	7	19	38	11	32	41	42	19	

2. Mengurutkan individu berdasarkan nilai *fitness*-nya dari yang terbesar ke yang terkecil untuk mendapatkan individu terbaik. Pada Tabel 4.15 menunjukkan urutannya.

Tabel 4.15 Urutan Populasi Berdasarkan *Fitness*

Individu	Kromosom												Fitness
P1	15	40	38	9	54	42	23	16	27	28	51	32	20.3287
	24	11	35	49	29	5	19	38	16	52	40	54	
P3	38	51	21	27	49	14	20	18	30	32	46	4	19.2699
	2	23	26	7	19	38	18	11	32	41	42	19	
C3	15	40	27	38	9	54	42	23	16	28	51	32	18.5075
	24	16	11	35	49	29	5	19	38	52	40	54	
C4	38	51	21	27	32	49	14	20	18	30	46	4	18.4543
	18	2	23	26	7	19	38	11	32	41	42	19	
P2	53	45	23	19	29	24	35	40	49	38	39	52	18.1280
	2	9	51	28	15	25	17	31	26	29	34	20	
C2	53	45	23	19	29	24	35	40	49	38	39	52	17.3161
	24	11	35	49	29	5	19	38	16	52	40	54	

C1	15	40	38	9	54	42	23	16	27	28	51	32	17.1386
	2	9	51	28	15	25	17	31	26	29	34	20	

3. Memilih individu terbaik sejumlah populasi untuk digunakan pada generasi selanjutnya. Pada Tabel 4.16 menunjukkan individu yang akan melanjutkan ke generasi berikutnya.

Tabel 4.16 Individu Generasi Baru

Individu	Kromosom												Fitness
P1	15	40	38	9	54	42	23	16	27	28	51	32	20.3287
	24	11	35	49	29	5	19	38	16	52	40	54	
P3	38	51	21	27	49	14	20	18	30	32	46	4	19.2699
	2	23	26	7	19	38	18	11	32	41	42	19	
C3	15	40	27	38	9	54	42	23	16	28	51	32	18.5075
	24	16	11	35	49	29	5	19	38	52	40	54	

Pada Tabel 4.15 menunjukkan bahwa P1, P3, dan C3 akan digunakan pada generasi berikutnya dengan P1 sebagai individu terbaik. Rincian dari P1 sebagai individu terbaik adalah sebagai berikut:

Kromosom : 15 40 38 9 54 42 23 16 27 28 51 32 24 11 35 49 29
5 19 38 16 52 40 54

Indeks Bahan Makanan : 5 29 9 3 17 31 6 4 9 21 12 7 8 8 9 11 9 4 5 9 5 38 10
12

Total Harga : 22298,35

Variasi : 18

Penalti Gizi : 437,86

Fitness : 20,3287

4.3 Perancangan Antarmuka

Pada perancangan antarmuka ini menjelaskan tentang perancangan antarmuka atau *user-interface* dari aplikasi yang akan dibuat. Halaman yang terdapat pada aplikasi ini terdiri dari halaman awal berupa *input* data anak, perhitungan algoritme genetika, serta hasil optimasi.

4.3.1 Input Data Anak

Halaman awal adalah *input* data anak. Pada halaman ini digunakan untuk meng-*input*-kan data anak berupa jenis kelamin, usia, tinggi badan, berat badan, aktivitas, alergi dan hari yang bisa dilihat pada Gambar 4.16.

Optimasi Susunan Bahan Makanan bagi Anak Penderita ADHD Menggunakan Algoritme Genetika	
Data Anak	Data Anak <div>Jenis Kelamin <input type="text"/></div> <div>Usia <input type="text"/></div> <div>Tinggi Badan <input type="text"/></div> <div>Berat Badan <input type="text"/></div> <div>Aktivitas <input type="text"/></div> <div>Alergi <input type="text"/></div> <div>Hari <input type="text"/></div> <div>Submit</div>
Perhitungan Algen	
Hasil Optimasi	

Gambar 4.16 Input Data Anak

4.3.2 Menu Perhitungan Algoritme Genetika

Setelah melakukan *input* data anak adalah meng-*input*-kan parameter yang digunakan dalam perhitungan algoritme genetika yaitu jumlah populasi, banyak generasi, *crossover rate*, dan *mutation rate* seperti yang ditunjukkan pada Gambar 4.17.

Optimasi Susunan Bahan Makanan bagi Anak Penderita ADHD Menggunakan Algoritme Genetika	
Data Anak	<div>Jumlah Populasi <input type="text"/></div> <div>Banyak Generasi <input type="text"/></div> <div>Cr <input type="text"/></div> <div>Mr <input type="text"/></div> <div>Submit</div>
Perhitungan Algen	
Hasil Optimasi	

Setelah menekan tombol submit, maka selanjutnya akan ditampilkan hasil dari perhitungan algoritme genetika. Pada halaman ini ditampilkan parameter yang telah dimasukkan sebelumnya, inisialisasi populasi awal, *offspring* dari proses *crossover*, *offspring* dari proses *mutation*. Pada Gambar 4.18 menunjukkan tam

Gambar 4.17 Input Parameter Algoritme Genetika

Optimasi Susunan Bahan Makanan bagi Anak Penderita ADHD Menggunakan Algoritme Genetika	
Data Keluarga	<div>Parameter Algoritme Genetika</div> <div><input type="text"/></div>
Perhitungan Algen	
Hasil Optimasi	

Inisialisasi

Gambar 4.18 Perhitungan Algoritme Genetika

4.3.3 Menu Hasil Optimasi

Menu hasil optimasi menampilkan hasil dari proses seleksi, yaitu individu terbaik yang didapatkan dari generasi terakhir. Pada menu ini akan ditampilkan juga susunan bahan makanan, kebutuhan gizi yang terpenuhi dan total biaya yang dikeluarkan dari individu terbaik. Pada Gambar 4.19 menunjukkan menu hasil optimasi.

Optimasi Susunan Bahan Makanan bagi Anak Penderita ADHD Menggunakan Algoritme Genetika	
Data Anak	<div>Hasil Optimal</div> <div></div>
Perhitungan Algen	
Hasil Optimasi	

Gambar 4.19 Menu Hasil Optimasi

BAB 5 IMPLEMENTASI

Bab ini menjelaskan tentang implementasi sistem berdasarkan analisis kebutuhan dan perancangan yang telah dibuat. Pada bab ini membahas penjelasan dari implementasi algoritme genetika dan implementasi antarmuka sistem.

5.1 Implementasi Algoritme Genetika

Pada sub bab ini menjelaskan implementasi dari *flowchart* yang sudah dibuat menjadi Bahasa pemrograman yang bias dimengerti oleh komputer. Implementasi pada penelitian ini menggunakan pemrograman javascript.

5.1.1 Proseses Input

Pada *file* app.js terdiri dari beberapa pseudocode. *Source code* mengenai proses *input* awal dijelaskan pada Gambar 5.1.

	app.js
1	<code>\$("#target").submit(function(event) {</code>
2	<code>var usia = event.target["usia"].value;</code>
3	<code>var jkelamin = event.target["jkelamin"].value;</code>
4	<code>var tinggib = event.target["tinggib"].value;</code>
5	<code>var beratb = event.target["beratb"].value;</code>
6	<code>var aktivitas = event.target["aktivitas"].value;</code>
7	<code>var hari = event.target["hari"].value;</code>
8	<code>}</code>
9	<code>\$("#parameter").submit(function(event) {</code>
10	<code>var populasi = event.target["populasi"].value;</code>
11	<code>var generasi = event.target["generasi"].value;</code>
12	<code>var cr = event.target["cr"].value;</code>
13	<code>var mr = event.target["mr"].value;</code>
14	<code>}</code>

Gambar 5.1 Source Code Proseses Input

Penjelasan Gambar 5.1 *source code* proses *input* sebagai berikut:

1. Pada baris 1 mendeklarasikan fungsi *target*
2. Pada baris 2 mendeklarasikan variabel *usia* dengan nilai sesuai dengan data *usia*
3. Pada baris 3 mendeklarasikan variabel *jkelamin* dengan nilai sesuai dengan data *jkelamin*
4. Pada baris 4 mendeklarasikan variabel *tinggib* dengan nilai sesuai dengan data *tinggib*
5. Pada baris 5 mendeklarasikan variabel *beratb* dengan nilai sesuai dengan data *beratb*

6. Pada baris 6 mendeklarasikan variabel *aktivitas* dengan nilai sesuai dengan data *aktivitas*
7. Pada baris 7 mendeklarasikan variabel *hari* dengan nilai sesuai dengan data *hari*
8. Pada baris 9 mendeklarasikan fungsi parameter
9. Pada baris 10 mendeklarasikan variabel *populasi* dengan nilai sesuai dengan data *populasi*
10. Pada baris 11 mendeklarasikan variabel *generasi* dengan nilai sesuai dengan data *generasi*
11. Pada baris 12 mendeklarasikan variabel *cr* dengan nilai sesuai dengan data *cr*
12. Pada baris 13 mendeklarasikan variabel *mr* dengan nilai sesuai dengan data *mr*

5.1.2 Proses Representasi Kromosom

Pada proses representasi kromosom ditentukan dengan nilai acak yang dikalikan dengan permutasi 1 sampai dengan 55. Panjang kromosom didapatkan dengan mengalikan hari dengan 12. Jumlah kromosom awal dibuat sejumlah populasi.

	app.js
1	var jkromosom = hari * 12;
2	var parent = new Array(populasi);
3	for (var i = 0; i < populasi; i++) {
4	parent[i] = new Array(jkromosom);
5	for (var j = 0; j < jkromosom; j++) {
6	parent[i][j] = Math.floor(Math.random() * 55) +
7	1;
8	}
9	console.log(parent[i]); }

Gambar 5.2 Source Code Representasi Kromosom

Penjelasan Gambar 5.2 *source code* representasi kromosom sebagai berikut:

1. Pada baris 1 mendeklarasikan variable *jkromosom* dengan nilai hasil dari perkalian antara hari dengan 12
2. Pada baris 2 mendeklarasikan variabel *parent* sebagai *array* dengan panjang *populasi*
3. Pada baris 3 melakukan perulangan hingga persyaratan *i < populasi* tidak terpenuhi dimana nilai *i* terus bertambah satu setiap perulangan dilakukan
4. Pada baris 4 mendeklarasikan variabel *parent[i]* sebagai *array* dengan panjang *jkromosom*
5. Pada baris 5 melakukan perulangan hingga persyaratan *j < jkromosom* tidak terpenuhi dimana nilai *j* terus bertambah satu setiap perulangan dilakukan

6. Pada baris 6 memberikan nilai pada *parent[i][j]* berupa hasil dari nilai acak dikalikan dengan 55 yang hasilnya dijumlahkan dengan 1
7. Pada baris 9 menampilkan isi dari *parent[i]*

5.1.3 Crossover

	app.js
1	var parentcr = []
2	parentcr[-1] = -1;
3	for (var i = 0; i < jcrossover;) {
4	parentcr[i] = Math.floor(Math.random() * populasi);
5	if (parentcr[i] === parentcr[i-1]) {
6	i--
7	} else {
8	
9	}
10	i++
11	}
12	
13	for (var i = 0; i < jcrossover;) {
14	var partisikromosom =
15	_.partition(parent[parentcr[i]], function(row, index){
16	return index < (parent[parentcr[i]].length/2);
17	})
18	var partisikromosom2 =
19	_.partition(parent[parentcr[i+1]], function(row,
20	index){
21	return index < (parent[parentcr[i+1]].length/2);
22	})
23	parent[parent.length] =
24	partisikromosom[0].concat(partisikromosom2[1])
25	parent[parent.length] =
26	partisikromosom2[0].concat(partisikromosom[1])
27	i+=2;
28	}
29	if (Math.round(populasi * cr) % 2 !== 0) {
30	parent.splice(parent.length-1,1)
31	}

Gambar 5.3 Source Code Crossover

Penjelasan Gambar 5.3 *source code crossover* sebagai berikut:

1. Para baris 1 mendeklarasikan variabel *parentcr* sebagai *array*
2. Pada baris 2 memberikan nilai -1 pada *parentcr* indeks -1. Hal ini bertujuan untuk menghindari kesalahan dalam perulangan pada *source code* baris ke 3
3. Pada baris 3 melakukan perulangan hingga persyaratan *i < jcrossover* tidak terpenuhi
4. Pada baris 4 memberikan nilai pada *array parentcr* indeks ke *i* berupa hasil dari nilai acak dikalikan dengan *populasi*
5. Pada baris ke 5 melakukan pengecekan nilai dari *parentcr* indeks ke *i* tidak sama dengan *parentcr* indeks ke *i - 1*. Jika benar maka melakukan pengurangan nilai *i* sebanyak 1
6. Pada baris ke 10 melakukan penambahan nilai *i* sebanyak 1

7. Pada baris ke 13 melakukan perulangan hingga persyaratan $i < j_{crossover}$ tidak terpenuhi
8. Pada baris ke 14 mendeklarasikan variabel *partisikromosom* yang berisi partisi dari *array parent* pada indeks nilai partisicr indeks ke *i*
9. Pada baris ke 18 mendeklarasikan variabel *partisikromosom2* yang berisi partisi dari *array parent* pada indeks partisicr indeks ke *i+1*
10. Pada baris ke 23 memberikan nilai berupa gabungan dari *partisikromosom* indeks ke 0 dengan *partisikromosom2* indeks ke 1 pada *array parent* pada indeks sesuai dengan panjang *array parent*
11. Pada baris ke 25 memberikan nilai berupa gabungan dari *partisikromosom2* indeks ke 0 dengan *partisikromosom* indeks ke 1 pada *array parent* pada indeks sesuai dengan panjang *array parent*
12. Pada baris ke 27 melakukan penambahan nilai *i* sebanyak 2
13. Pada baris ke 29 melakukan pengecekan apabila hasil dari *populasi* dikalikan dengan *cr* lalu di mod 2 tidak sama dengan 0, maka menghapus indeks terakhir dari *array parent*

5.1.4 Mutasi

	app.js
1	function partisi(items, size) {
2	var result = _.groupBy(items, function(item, i) {
3	return Math.floor(i/size);
4	});
5	return _.map(result, function(val, key) {
6	return val;
7	});
8	}
9	
10	
11	for (var i = 0; i < jmutasi;i++) {
12	partisimr = partisi(parent[parentmr[i]], 12)
13	for (var j = 0; j < partisimr.length;j++) {
14	mutation[j] = swapArrayElements(partisimr[j],
15	Math.floor(Math.random() * 6), Math.floor(Math.random()
16	* 12))
17	console.log(mutation[j]);
18	}
19	if (partisimr.length === 1){
20	parent[parent.length] = mutation[0];
21	}
22	else if (partisimr.length === 2){
23	parent[parent.length] =
24	mutation[0].concat(mutation[1]);
25	}
26	else if (partisimr.length === 3){
27	parent[parent.length] =
28	mutation[0].concat(mutation[1],mutation[2]);
29	}
30	else if (partisimr.length === 4){
31	parent[parent.length] =
32	mutation[0].concat(mutation[1],mutation[2],mutation[3])
33	;
34	}
35	else if (partisimr.length === 5){

```

36     parent[parent.length] =
37     mutation[0].concat(mutation[1],mutation[2],mutation[3],
38     mutation[4]);
39     }
40     else if (partisimr.length === 6){
41         parent[parent.length] =
42         mutation[0].concat(mutation[1],mutation[2],mutation[3],
43         mutation[4],mutation[5]);
44     }
45     else {
46         parent[parent.length] =
47         mutation[0].concat(mutation[1],mutation[2],mutation[3],
48         mutation[4],mutation[5],mutation[6]);
49     }

```

Gambar 5.4 Source Code Mutasi

Penjelasan Gambar 5.4 *source code* mutasi:

1. Para baris 1 hingga 8 mendeklarasikan fungsi partisi yang digunakan dalam pemotongan kromosom menjadi beberapa bagian
2. Pada baris 11 melakukan perulangan hingga persyaratan $i < j_{mutasi}$ tidak terpenuhi
3. Pada baris 12 mendeklarasikan variabel *partisimr* yang berisi hasil dari menjalankan fungsi partisi
4. Pada baris 13 melakukan perulangan hingga persyaratan $j < partisimr.length$ tidak terpenuhi
5. Pada baris 14 mendeklarasikan *array mutation* yang berisi hasil dari fungsi *swapArrayElements*
6. Pada baris 19 jika *partisimr.length* sama dengan 1, maka *parent[parent.length]* diisi *mutation[0]*
7. Pada baris 22 jika *partisimr.length* sama dengan 2, maka *parent[parent.length]* diisi penggabungan isi *mutation[0]* dan *mutation [1]*
8. Pada baris 26 jika *partisimr.length* sama dengan 3, maka *parent[parent.length]* diisi penggabungan isi *mutation[0]* sampai *mutation [2]*
9. Pada baris 30 jika *partisimr.length* sama dengan 4, maka *parent[parent.length]* diisi penggabungan isi *mutation[0]* sampai *mutation [3]*
10. Pada baris 35 jika *partisimr.length* sama dengan 5, maka *parent[parent.length]* diisi penggabungan isi *mutation[0]* sampai *mutation [4]*
11. Pada baris 40 jika *partisimr.length* sama dengan 6, maka *parent[parent.length]* diisi penggabungan isi *mutation[0]* sampai *mutation [5]*
12. Pada baris 45 jika *partisimr.length* diluar dari kondisi yang ditentukan, maka *parent[parent.length]* diisi penggabungan isi *mutation[0]* sampai *mutation [6]*

5.2 Implementasi Antarmuka

Agar pengguna dapat menggunakan sistem, maka diperlukan antarmuka sebagai perantara interaksi. Implementasi antarmuka sistem menggunakan satu halaman yang dibagi menjadi tiga bagian utama. Tampilan pada bagian pertama sistem yang berisi *input* data anak ditunjukkan pada Gambar 5.5.

The screenshot shows a web browser window with the address bar displaying 'localhost/dashboard/index.html'. The page title is 'Optimasi Susunan Bahan Makanan untuk Anak ADHD'. The left sidebar contains three menu items: 'Data Anak' (selected), 'Parameter', and 'Hasil Optimasi'. The main content area is titled 'Data' and contains the 'Input Data Anak' form. The form fields are as follows:

Field	Value
USIA	2 tahun
JENIS KELAMIN	Laki-laki
BERAT BADAN	dalam satuan kg
TINGGI BADAN	dalam satuan cm
AKTIVITAS	Sangat Ringan
ALERGI	
HARI	1

A 'PROSES' button is located below the form fields.

Gambar 5.5 Tampilan Bagian Pertama *Input* Data Anak

Pada Gambar 5.5 dijelaskan bahwa pada halaman pertama pengguna diharuskan melakukan *input* data anak berupa usia, jenis kelamin, berat badan, tinggi badan, aktivitas, alergi, dan jumlah hari yang diinginkan. Setelah seluruh data diisi, maka pengguna dapat menekan tombol proses agar data yang dimasukkan dapat di proses.

Setelah *input* data anak berhasil di proses, selanjutnya pengguna diharuskan melakukan *input* parameter dari algoritme genetika. *Input* yang diperlukan pada bagian kedua ini berupa jumlah populasi, jumlah generasi, *crossover rate* (*Cr*), dan *mutation rate* (*Mr*). Setelah parameter dimasukkan, maka pengguna dapat menekan tombol *submit* sehingga data yang dimasukkan dapat di proses dalam perhitungan algoritme genetika. Tampilan dari bagian kedua *input* parameter dapat dilihat pada Gambar 5.6.

Gambar 5.6 Tampilan Bagian Kedua *Input Parameter*

Setelah proses perhitungan algoritme genetika selesai maka akan ditampilkan hasilnya dapat dilihat pada Gambar 5.7. Pada tampilan kedua proses optimasi ditampilkan parameter algoritme genetika yang sudah dimasukkan, inisialisasi, *crossover*, dan mutasi.

Gambar 5.7 Tampilan Bagian Kedua Proses Optimasi

Pada bagian ketiga hasil optimasi ditampilkan hasil akhir dari proses algoritme genetika. Pada Gambar 5.8 ditunjukkan tampilan dari bagian ketiga hasil optimasi yang terdiri dari kromosom terbaik, susunan bahan makanan yang di kelompokkan berdasarkan hari, dan total harga keseluruhan bahan makanan. Pengguna dapat menggunakan kembali sistem dengan data masukkan yang baru dengan menekan tombol *refresh page* yang akan mengembalikan pengguna pada bagian pertama *input data anak*.

Kromosom	Representasi Kromosom
P1	12,1,42,19,28,8,33,7,49,45,14,22,6,11,21,13,43,50,40,30,46,23,31,37

HARI KE- 1	HARI KE- 2	
Makaroni, Cumi-cumi, Selai	Kentang, Ikan	15367.5
Kacang Tanah, Minyak Kacang	kembung, Kacang Tanah	
Kedele, Nasi Ketan Hitam, Ikan	Kupas, Minyak	
asin	Jagung, Singkong, Kuning	
kering, Oncom, Mayonnaise, Tape	telur ayam, Pete	
Beras Ketan, Usus sapi, Kacang	Segar, Mentega, Sukun, Putih	
Tanah, Minyak Zaitun	telur ayam, Kembang	
	Tahu, Kelapa	

Refresh Page

Gambar 5.8 Tampilan Bagian Ketiga Hasil Optimasi

BAB 7 PENUTUP

7.1 Kesimpulan

Sesuai dengan hasil implementasi dan pengujian optimasi susunan bahan makanan bagi anak penderita *Attention Deficit Hyperactivity Disorder* menggunakan metode Algoritme Genetika, dengan itu bisa diambil kesimpulan yaitu:

1. Algoritme Genetika dapat digunakan untuk pemecahan masalah optimasi susunan bahan makanan bagi anak penderita *Attention Deficit Hyperactivity Disorder* pada penelitian ini menggunakan representasi kromosom permutasi. Nilai-nilai gen pada kromosom dibangkitkan secara acak. Panjang kromosom ditentukan berdasarkan jumlah hari yang diinginkan, dimana pada satu harinya terdapat 12 gen. Metode *crossover* yang digunakan adalah *one-cut point*. Kemudian untuk metode mutasi menggunakan *exchange mutation*. Metode seleksi yang digunakan yaitu *elitism*. Nilai *fitness* menjadi patokan utama dalam menentukan solusi dimana semakin besar nilai *fitness*-nya, maka semakin baik solusi yang didapatkan. Hal ini berlaku juga pada nilai *fitness* yang semakin kecil, maka solusi yang didapatkan kurang baik. Nilai *fitness* didapatkan dari perhitungan penalti harga, penalti gizi, dan variasi bahan makanan.
2. Parameter-parameter Algoritme Genetika yang ada dalam penelitian ini yaitu jumlah populasi (*popSize*), *crossover rate* (*Cr*), *mutation rate* (*Mr*), dan jumlah generasi. Pada penelitian ini pengujian yang dilakukan adalah pengujian jumlah populasi, pengujian kombinasi *Cr* dan *Mr*, dan pengujian jumlah generasi. Setiap pengujian dilakukan sebanyak 10 kali. Pengujian jumlah populasi menggunakan *Cr* 0.4, *Mr* 0.2, dan 1 generasi dihasilkan nilai rata-rata *fitness* terbesar berada pada jumlah populasi sebanyak 100 dengan nilai rata-rata *fitness* 13.690. Nilai rata-rata *fitness* tertinggi didapatkan pada jumlah populasi 100 dikarenakan semakin banyak populasinya, maka variasi semakin tinggi yang berpengaruh besar pada nilai *fitness* yang didapatkan. Pada pengujian kombinasi *Cr* dan *Mr* dengan 6 populasi dan 2 generasi didapatkan nilai rata-rata *fitness* tertinggi dicapai pada kombinasi *Cr* dan *Mr* 0.3 dan 0.7 dengan nilai rata-rata *fitness* 13.594. Pada pengujian terakhir yaitu pengujian jumlah generasi dengan 20 populasi, *Cr* 0.6, dan *Mr* 0.2 didapatkan nilai rata-rata *fitness* tertinggi 13.928 pada jumlah generasi 50. Ditemukan terjadinya konvergensi dini sehingga nilai *fitness* pada generasi selanjutnya tetap.

7.2 Saran

Dari hasil penelitian yang didapatkan, berikut beberapa saran yang bisa digunakan dalam mengembangkan penelitian ini di masa yang akan datang:

1. Disarankan pada penelitian selanjutnya menggunakan metode *crossover* dan mutasi yang berbeda dengan penelitian ini. Penggunaan metode *crossover one-cut point* dan metode mutase *exchange mutation* terkadang kurang optimal karena menyebabkan terjadinya konvergensi dini.
2. Pada implementasi dan pengujian aplikasi memakan waktu yang cukup lama untuk mendapatkan hasil optimasinya. Oleh karena itu penulis menyarankan kedepannya agar dilakukan pengembangan aplikasi yang dapat bekerja lebih cepat, misalkan dengan menggunakan proses parallel berbasis GPU.



DAFTAR PUSTAKA

- Almatsier, S. 2009. *Prinsip Dasar Ilmu Gizi*. Jakarta: Gramedia Pustaka Utama.
- American Psychiatric Associations. 2013. Diagnostic and Statistical Manual of Mental Disorders fifth edition (DSM-V). [e-book] Washington DC: American Psychiatric Associations. Tersedia di: <<http://psygradaran.narod.ru/lib/clinical/DSM5.pdf> > [Diakses 20 Februari 2016]
- Panhares, YG & Mahmudy, WF. 2015. *Optimasi Distribusi Barang dengan Algoritme Genetika*, DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya, vol. 5, no. 11.
- Hartono, A. 2006. *Terapi Gizi dan Diet Rumah Sakit (2nd ed.)*. Jakarta: Buku Kedokteran EGC.
- Mahmudy, W. F. (2016, Maret 30). *Wayan Firdaus Mahmudy*. Retrieved Agustus 30, 2016, from Modul Algoritme Evolusi – Semester Ganjil 2015-2016: <http://wayanfm.lecture.ub.ac.id/files/2016/03/2015-Algoritme-Evolusi-Modul.pdf>
- Arif, C, Setiawan, BI, Mizoguchi, M. 2014. *Penentuan Kelembaban Tanah Optimum Untuk Budidaya Padi Sawah SRI (System of Rice Intensification) Menggunakan Algoritme Genetika*. Jurnal Irigasi – Vol.9, No.1.
- Jacobson, M. F. & Schardt, D. 1999, *Diet, ADHD & Behaviour*, online diakses 7 Oktober 2016 (http://www.adhd.co.il/pdf/adhd_resch_bk02.pdf)
- Suci, WW, Mahmudy, WF & Putri, RRM. 2015. *Optimasi Biaya Pemenuhan Gizi dan Nutrisi pada Manusia Lanjut Usia Menggunakan Algoritme Genetika*. DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya, vol. 5, no. 17.